UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

Departamento de Informática y Automática

TESIS DOCTORAL

Integración de técnicas de procesamiento del lenguaje natural para la recuperación de información en bibliotecas de componentes software

TESIS DOCTORAL

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Manuel de Buenaga Rodríguez

Director:

Alfredo Fernández-Valmayor Crespo

Madrid, 2001

ISBN: 978-84-669-0389-9

INTEGRACION DE TECNICAS DE PROCESAMIENTO DEL LENGUAJE NATURAL PARA LA RECUPERACION DE INFORMACION EN BIBLIOTECAS DE COMPONENTES SOFTWARE

Memoria que presenta para optar al grado de Doctor en Ciencias Físicas

Manuel de Buenaga Rodriguez

Dirigida por el profesor

Alfredo Fernández-Valmayor Crespo

Departamento de Informática y Automática
Facultad de Ciencias Físicas
Universidad Complutense de Madrid

Enero 1996

A Ester

AGRADECIMIENTOS [*]

Gracias a todos los que en tan gran medida me han ayudado en el desarrollo de esta tesis.

Mi más profundo agradecimiento a Alfredo Fernández-Valmayor, mi director de tesis. A Alfredo debo mi introducción en el campo del Procesamiento del Lenguaje Natural. Su conocimiento, dedicación y paciencia han sido la pieza clave en que este trabajo haya llegado a buen término. A Carmen Fernández Chamizo por su constante consejo y su visión crítica, inestimablemente constructiva. A Antonio Vaquero, Manuel Ortega y Luis Hernández por su apoyo en todo momento.

Gracias a Baltasar Fernández Manjón y a Pedro González Calero. Sus aportaciones a este trabajo han sido realmente innumerables. Ellos han contribuido con ideas técnicas, teóricas y de toda índole, así como con parte importante de las ingentes dosis de ánimo que han sido necesarias en muchos de los momentos difíciles. A Juan Lanchares, Fernando Trescastro, José María Gómez Hidalgo, Mercedes Gómez Albarrán, Ana Fernández-Pampillón y a todos los compañeros del grupo de trabajo del Departamento de Informática y Automática que, con un ambiente inmejorable, han contribuido al desarrollo de esta investigación.

Gracias a mi familia y todos los amigos de los que he recibido ilusión y aliento a lo largo de todos estos años de trabajo.

Finalmente, gracias a ti Ester, por tu cariño y comprensión, sin ellos, todo esto, como tantas otras cosas, no habría sido posible.

[*]Esta investigación ha sido mantenida parcialmente gracias a una beca del Plan Nacional de Formación del Personal Investigador (FPI 90/92) y dos proyectos de investigación financiados por la Comisión Interministerial de Ciencia y Tecnología (CICYT TIC92-0058 y TIC94-0187)

INDICE

CAPITULO 1

INTRODUCCION

La gran mayoría del conocimiento humano se encuentra almacenado en forma de texto. El desarrollo de técnicas para almacenar y buscar documentos es casi tan antiguo como el mismo lenguaje escrito. Sin embargo, los sistemas informáticos han cambiado la forma en que se almacenan, buscan y recuperan los documentos. La cantidad de información disponible en la actualidad en formato electrónico es tremenda y su ritmo de crecimiento aumenta día a día. El desarrollo de las redes de computadoras y los medios de almacenamiento masivo de datos durante los últimos años ha provocado una aceleración de esta tendencia.

Los sistemas de Recuperación de Información (RI) tienen como finalidad esencial facilitar el acceso a la información, seleccionando entre grandes conjuntos de documentos aquellos adecuados a las necesidades del usuario. El contenido textual de los documentos es procesado en los sistemas de RI para obtener una representación interna. Esta representación interna es utilizada posteriormente para proporcionar los documentos más adecuados a las consultas formuladas por los usuarios. Desde hace más de dos décadas, la mayoría de los modelos orientados a la implementación de este tipo de sistemas utilizan criterios estadísticos, basados fundamentalmente en la frecuencia de aparición de los términos en los documentos, para realizar el análisis de los textos. En este sentido, en la actualidad existe una serie de técnicas que proporcionan las bases para el desarrollo de sistemas de RI prácticos, plenamente funcionales, y capaces de implementar el proceso de recuperación con un comportamiento difícil de mejorar por aproximaciones más sofisticadas.

Por otra parte, las técnicas utilizadas para el desarrollo de los sistemas basados en el Procesamiento del Lenguaje Natural (PLN) son muy diferentes de las anteriores, y en ellas el análisis sintáctico, semántico y contextual juega un papel central. La investigación desarrollada en torno al PLN tiene como objetivo la consecución de sistemas informáticos que implementen funcionalidades próximas a la comprensión y la generación del lenguaje humano. Aplicaciones típicas que utilizan estas técnicas son los sistemas de traducción automática, interfaces en lenguaje natural a sistemas informáticos (e.g. bases de datos, sistemas operativos y sistemas expertos) y sistemas de extracción de información. La naturaleza del problema de la

comprensión del lenguaje natural, también enmarcado frecuentemente dentro de la Inteligencia Artificial, ha hecho que hasta tiempos recientes, la mayoría de los sistemas de este tipo adolecieran de una constante condición de prototipo, presentando implementadas sus funcionalidades sólo parcialmente, o procesando subconjuntos del lenguaje natural excesivamente limitados como para resultar verdaderamente práctica su aplicación. No obstante, durante los últimos años se ha producido una evolución importante en el área. En la actualidad existe una serie de aplicaciones del PLN para las que resulta factible el desarrollo de sistemas que implementan funciones de forma apta para su utilización práctica sobre dominios concretos (e.g. noticias económicas, informes técnicos). Por otra parte, estos sistemas necesitan utilizar una importante cantidad de conocimiento específico del dominio y su coste de desarrollo es importante.

Desde el comienzo de las investigaciones en ambos campos, la integración de técnicas de PLN en la RI se ha considerado como un elemento decisivo para la consecución de mejoras significativas en el proceso de recuperación. Resulta plausible admitir que el procesamiento de los textos que realizan los sistemas de RI puede beneficiarse de la utilización de técnicas que llevan a cabo una interpretación más profunda del lenguaje que la basada exclusivamente en criterios estadísticos. También puede decirse lo mismo del procesamiento de las solicitudes de los usuarios que se realizan en lenguaje natural en sistemas de RI. Sin embargo, durante mucho tiempo la investigación en ambos campos ha discurrido por caminos divergentes. El estado de los modelos teóricos y las técnicas concretas desarrolladas en el campo PLN no era tal que permitiese la consecución de sistemas capaces de procesar conjuntos de documentos de dimensiones tales como las necesitadas por los sistemas de RI. De hecho, durante mucho tiempo, las investigaciones desarrolladas en esta dirección integradora han obtenido resultados negativos.

En los últimos años, y en consonancia con el desarrollo experimentado en el campo del PLN, existe un importante y renovado esfuerzo investigador encaminado a la integración de este tipo de técnicas en la RI. En la actualidad, existen diversas propuestas de modelos y sistemas que presentan diferentes problemas, pero que han conseguido mejoras en la efectividad, a diferencia de los resultados obtenidos en tiempos pasados. A pesar de esto, los trabajos dirigidos en este sentido aún son propuestas aisladas y se hacen necesarios, cada vez más, estudios integradores que permitan la comparación de las diferentes aproximaciones.

En esta tesis estudiamos la integración de técnicas de PLN en la RI y presentamos un modelo para su utilización en un dominio concreto: la documentación en lenguaje natural existente en los entornos de desarrollo de los sistemas informáticos, y más concretamente en las bibliotecas de componentes software. La documentación en formato electrónico que incluyen estos entornos puede ser utilizada por sistemas de ayuda orientados a facilitar su utilización por parte de los usuarios. Para demostrar la viabilidad de nuestro modelo, hemos desarrollado dos sistemas, Argos y Ares. Ambos sistemas se han desarrollado para una biblioteca de componentes con documentación en lenguaje natural específica, aunque la aproximación seguida en ambos resulta de aplicación inmediata a otras bibliotecas diferentes. En concreto, los dos sistemas han sido desarrollados para el conjunto de órdenes que proporciona

el sistema operativo UNIX y la documentación en lenguaje natural asociada a ellos, en formato electrónico, que incluye este entorno.

El sistema Argos implementa un amplio conjunto de funcionalidades que permiten proporcionar ayuda al usuario del sistema operativo UNIX. En la definición del sistema proponemos un modelo de sistema de ayuda a la utilización de bibliotecas de componentes software basado en la documentación. En él juega un papel central el procesamiento, tanto del texto del manual del sistema operativo, como de las necesidades del usuario también expresadas en lenguaje natural. Argos incluye para la implementación de las funciones de análisis de textos y consultas, un módulo basado en el modelo del espacio vectorial y se fundamenta en criterios esencialmente estadísticos.

El sistema Ares se centra en la utilización de técnicas de PLN para obtener mejoras respecto a la aproximación seguida en Argos. Ares es un sistema totalmente implementado que constituye la definición de un modelo de sistema de RI basado en el PLN plenamente factible, orientado a conseguir mejoras en la efectividad del proceso de recuperación y reducir inconvenientes que plantean los sistemas de este tipo.

El orden de exposición que seguimos en el desarrollo de esta memoria es el siguiente:

En el capítulo 2 se analiza el problema de la Recuperación de la Información, concebido como un proceso cuyos aspectos fundamentales son, tanto el análisis de los documentos textuales, como el de las solicitudes de información de los usuarios en lenguaje natural. Se presentan todos los aspectos relacionados con la efectividad del proceso de recuperación. A continuación se tratan las aproximaciones basadas en criterios estadísticos y con especial detalle el modelo del espacio vectorial. Después se estudian los elementos adicionales relacionados con la indexación automática de los documentos y consultas que sirven de base para el desarrollo de los sistemas de RI.

En el capítulo 3 se estudian las formas en que las técnicas de RI pueden facilitar la utilización de bibliotecas de componentes software. Como resultado de este estudio se presenta el diseño y construcción del sistema Argos, detallándose sus objetivos, funcionalidades y arquitectura. El sistema Argos se compone de diversos módulos (e.g. interfaz, modelo de usuario) entre los destaca el módulo de Recuperación de Información. De este módulo se detallan sus principales elementos: las técnicas de análisis de documentos y consultas utilizadas, y su arquitectura.

En el capítulo 4 se estudian los aspectos del Procesamiento del Lenguaje Natural más directamente relacionados con el desarrollo de sistemas orientados a la Recuperación de Información. Se introducen los principales niveles de descripción lingüística. A continuación se estudian las grandes bases de conocimiento y las gramáticas de unificación (elementos que juegan un papel importante en el sistema Ares). Se tratan los sistemas que implementan funcionalidades más próximas a las de la Recuperación de Información: interfaces en lenguaje natural, sistemas de comprensión de textos y sistemas de extracción de información. Después se estudian

los sistemas de Recuperación de Información basados en la utilización de técnicas de PLN distinguiéndose dos aproximaciones: la basada en la sintaxis y la basada en la semántica. Una conclusión global de este estudio es que los sistemas enmarcados en la aproximación semántica proporcionan mejoras más importantes en la efectividad del proceso de recuperación, pero resultan excesivamente dependientes del dominio y conllevan un importante esfuerzo de desarrollo.

En el capítulo 5 se presenta el sistema Ares. Este sistema materializa nuestro modelo para la integración de técnicas de PLN en la RI en el dominio concreto de las bibliotecas de componentes software. Comenzamos analizando las características del dominio concreto en el que se centra el sistema, destacando el problema de la brevedad de las descripciones en lenguaje natural incluidas en colecciones de componentes software. A continuación se detalla el planteamiento y la organización del sistema: se sigue una aproximación cercana a la de los sistemas basados en la semántica, con vistas a introducir mejoras en la efectividad del proceso de recuperación, pero reduciendo al mismo tiempo el esfuerzo de desarrollo. Para reducir el esfuerzo de desarrollo se realiza la construcción automática del léxico a partir de información existente en una base de datos léxica, WordNet y se utiliza una gramática de unificación para la implementación del analizador-traductor. Se tratan los principales aspectos del sistema: la construcción del léxico, la representación de las descripciones y consultas, la definición de la gramática orientada al análisis de las expresiones en lenguaje natural y la forma en que se calcula la similitud entre documentos y consultas. Finalmente se presentan una serie de experimentos encaminados al estudio comparativo de aspectos del sistema centrados en la efectividad.

En el capítulo 6 resumimos las principales aportaciones realizadas y enumeramos las posibles líneas de desarrollo en un futuro inmediato y a más largo plazo. Se incluye un apéndice final con los detalles de implementación del sistema Ares y datos experimentales, seguido de la bibliografía utilizada para el desarrollo de esta memoria.

CAPITULO 2

LA RECUPERACION DE LA INFORMACION

2.1 Introducción

En tiempos pasados la cantidad total de información disponible variaba de forma relativamente lenta, en la actualidad la tasa de crecimiento del conocimiento existente en forma de texto se ha incrementado de forma espectacular. Como datos concretos y a modo de ilustración, podemos decir que de 1800 a 1970, el número de publicaciones científicas pasó de 100 a 100.000, y que en la década de 1970 a 1980 se pusieron en circulación anualmente cerca de 2.000.000 de escritos, lo que equivale a unos 7.000 informes diarios. Durante la década pasada, el ritmo de introducción de nuevos ítems en la Biblioteca del Congreso de EE.UU. llegó a ser de 3.500 diarios. En la actualidad parece no existir límite en el ritmo de crecimiento de la información [Coll-Vinent80, Salton89a]. La utilización de los sistemas informáticos ha sido el principal soporte de esta tendencia durante las últimas décadas. La evolución experimentada por los medios de almacenamiento de datos masivos y las redes de computadoras hace que la información accesible hoy en día a un usuario medio sea mayor que nunca. La información existente en bases de datos documentales accesibles por estos medios abarca en la actualidad un amplio rango de tipos; ejemplos concretos son periódicos, informes bursátiles, noticias económicas, resúmenes o textos completos de artículos de revistas científicas, e informes técnicos procedentes de departamentos de universidades [Williams93, Krol93, Hahn95, Buenaga95, Fernández-Manjón95c].

La investigación desarrollada en el campo de la Recuperación de Información (*Information Retrieval*) se centra en el desarrollo de modelos y sistemas informáticos que faciliten la selección en grandes colecciones de documentos de aquellos que son relevantes a necesidades concretas de los usuarios [Rijsbergen79, Salton83]. Los sistemas de Recuperación de Información (RI) procesan el contenido textual de los documentos de una colección, para obtener un representación interna de ellos orientada a facilitar la selección de los ítems adecuados a las necesidades concretas de los usuarios.

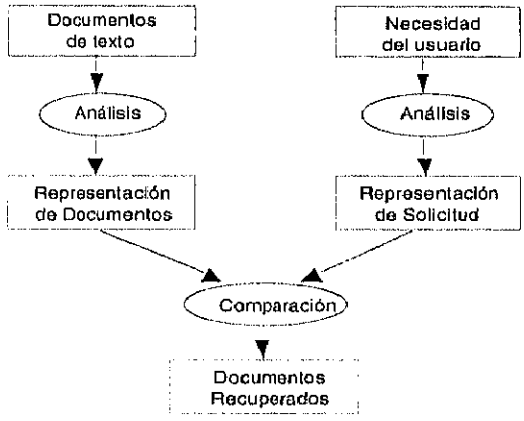


Figura 2.1: El proceso de recuperación de información.

En este capítulo describimos el marco en que se desarrolla el proceso de recuperación de la información, así como los modelos teóricos y las técnicas orientadas al desarrollo de este tipo de sistemas. En los puntos siguientes destacamos en primer lugar los aspectos del proceso en los que el análisis de documentos y consultas en lenguaje natural juega un papel importante, se detallan a continuación los aspectos relacionados con la efectividad de estos sistemas, para después pasar a tratar el modelo del espacio vectorial, y las técnicas de indexación automática de documentos y consultas. El capítulo termina con un resumen y conclusiones.

2.2 El proceso de Recuperación de la Información

Un sistema de Recuperación de Información está orientado al procesamiento de dos fuentes de información: un conjunto de documentos en forma de texto y las solicitudes de los usuarios. El sistema debe ser capaz de diferenciar entre el conjunto total de los documentos, aquellos que se adecuan a la solicitud del usuario y aquellos que no, seleccionando los primeros como relevantes. Los documentos relevantes se ofrecen al usuario como documentos recuperados. En la figura 2.1 podemos ver esquematizado este proceso [Croft93].

En la bibliografía [Rijsbergen79, Salton83a, López-Muñiz84, Salton89a] se pueden encontrar descritas diferentes formas de realizar este proceso: manual, automática o semiautomática. En nuestro trabajo nos centramos en el proceso realizado de forma totalmente automática y suponiendo que tanto los textos como las consultas a se encuentran expresados originalmente en lenguaje natural.

El proceso más general de recuperación de información puede descomponerse en tres subprocesos. Estos son: la indexación de los documentos, centrada en el análisis y representación de los documentos; los métodos de formulación y análisis de las

consultas de los usuarios; y el cálculo de la similitud entre documentos y consultas, que permite la comparación y la selección de aquellos documentos relevantes a las necesidades del usuario. Como veremos, estos tres subprocesos se encuentran fuertemente interrelacionados.

2.2.1 La indexación de los documentos

El proceso mediante el que se obtiene una representación interna de los documentos se suele denominar indexación [Salton83a, López-Muñiz84, Croft93]. Para la implementación del proceso de indexación es necesario definir primero una forma de representación (o lenguaje de representación) de la información contenida en los documentos y después desarrollar el analizador que, tomando como entrada el texto de los documentos, obtenga la representación de los mismos.

Con vistas a su adecuación al tratamiento informatizado de los documentos se han presentado diferentes formas de representación de los documentos como alternativas a la clasificación enumerativa o jerárquica convencional (i.e. la Clasificación Decimal Universal (CDU) o Dewey, utilizada clásicamente en la gestión documental) [Salton83a, Lopez-Muñiz84, Cleverdon91]. Las dos más difundidas han sido la representación basada en facetas y la basada en términos o palabras clave [Cleverdon91].

La representación basada en facetas propone la representación del contenido de los documentos mediante un conjunto de atributos o facetas predeterminados, a los que se les asigna valores específicos de acuerdo con la información existente en el documento. Este tipo de representación facilita la gestión de los documentos mediante sistemas de bases de datos con registros y campos, correspondientes a las facetas [Salton83a, López-Muñiz84]. En esta aproximación, el proceso de análisis de los documentos y asignación de valores a las facetas se realiza habitualmente de forma manual [Cleverdon91].

La representación basada en términos o palabras clave, propone la representación del contenido de un documento mediante una lista de términos o descriptores. Este tipo de representación se presta a que el análisis de los documentos para la obtención de su representación se pueda realizar de forma automática. La indexación basada en términos se puede hacer mediante vocabulario controlado o vocabulario no controlado. La utilización de vocabulario controlado supone la definición de un conjunto de términos fijo que son los únicos que pueden ser utilizados para la representación de los documentos, restricción que no es impuesta en el caso alternativo [Salton83a, Cleverdon91].

2.2.2 La formulación de las consultas y su procesamiento

Los lenguajes de consulta de bases de datos documentales han experimentado una evolución tendente a la simplificación de su uso por parte de los usuarios [Anick91,

TMC91, Hearst94]. De esta forma, los sistemas de los años 70 y principio de los 80, como DIALOG, se basaban en el uso de un lenguaje formal para la expresión de consultas [Salton83, Hearst94], con órdenes como *select* y operadores booleanos como *and* y *or*, y operadores de adyacencia o proximidad como *adj*, entre otros. Una búsqueda de documentos, como, por ejemplo, la de los que contengan a la vez los términos "information" y "retrieval" se realizaría de la siguiente forma en DIALOG:

```
SELECT INFORMATION
SELECT RETRIEVAL
COMBINE 1 AND 2
```

En contraste, la tendencia actual es a la utilización del lenguaje natural como medio para la formulación de las consultas. Sistemas actuales de Recuperación de Información como WAIS (Wide Area Information Service) [TMC91], explotan los avances desarrollados en el área de la *Interacción Hombre-Máquina* (*Human Computer Interaction* [Maddix90]) para el desarrollo del interfaz. En ellos el uso de menús desplegables, iconos y dispositivos señalizadores tipo ratón, constituyen la base para la formulación de órdenes, mientras que la especificación de la necesidad del usuario se realiza mediante el lenguaje natural. De esta forma, en este tipo de sistemas una consulta equivalente a la anterior sería especificada por el usuario como:

```
INFORMATION RETRIEVAL
```

Normalmente, el usuario puede introducir expresiones de mayor longitud que caracterizan en mayor detalle sus necesidades. De esta forma, otro ejemplo de consulta sería la siguiente:

```
AUTOMATIC INDEXING OF DOCUMENTS FOR INFORMATION RETRIEVAL IN
THE MEDICAL DOMAIN
```

El tipo de lenguaje en que el usuario puede expresar sus necesidades, o realizar sus consultas al sistema, se encuentra fuertemente relacionado con el método de representación de los documentos utilizado. De esta forma, en un sistema basado en facetas resulta especialmente difícil el procesamiento de consultas formuladas en lenguaje natural, e incluso puede resultar inadecuado [Bates83, Bates87]. En este estudio nos centraremos especialmente en modelos y técnicas orientados al procesamiento de consultas en lenguaje natural.

2.2.3 El cálculo de la similitud

El cálculo de la similitud entre documentos y consultas es el resultado del proceso de comparación representado en la figura 2.1, y tiene como finalidad la selección de los documentos que se adecuan a la solicitud del usuario [Salton83a, Croft93]. El valor de la similitud entre un documento y una consulta puede ser binario, entero, o real, dependiendo del modelo que se utilice para su cálculo [Harman92b].

En modelos que proporcionan un valor de la similitud binario, como el modelo booleano [Salton83a, Salton83b, Harman92b], la similitud de los documentos con una consulta permite la división del conjunto total de documentos en dos subconjuntos: el conjunto de los documentos no relevantes, con similitud 0, y el conjunto de los documentos relevantes, con similitud 1.

En modelos que proporcionan un valor de la similitud real (o entero), como el vectorial [Salton75, Salton83a] o el booleano extendido [Salton83b, Fox92b], resulta posible ofrecer al usuario los documentos relevantes ordenados por su similitud. La mayoría de los modelos y sistemas actuales incluyen esta ordenación por relevancia (ranking) en el proceso de recuperación [Harman92b].

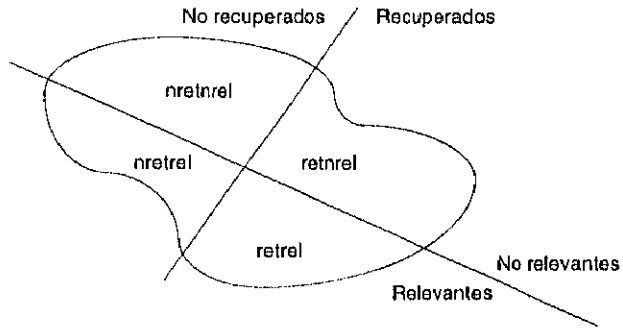
2.3 La efectividad del proceso de recuperación

En la bibliografía [Rijsbergen79, Salton83a, Bollman83, Raghavan89, Salton91b, Hull93] se puede encontrar un gran número de aspectos y parámetros utilizados para caracterizar el comportamiento de los sistemas de recuperación. Disponer de criterios para la evaluación del proceso de recuperación es una cuestión determinante a la hora de enjuiciar un determinado modelo o sistema concreto, o a la hora de comparar varias propuestas. También resultan de interés para determinar en qué forma podemos alterar el funcionamiento de un sistema para que el resultado del proceso de recuperación que implementa (i.e. los documentos recuperados) se adapte mejor a unas necesidades concretas (e.g. tiempo de búsqueda, número total de documentos recuperados, etc.)

En la evaluación de un sistema se pueden diferenciar aspectos relacionados con la *efectividad* y aspectos relacionados con la *eficiencia* [Salton83a, Salton89a]. Los aspectos relacionados con la eficiencia se centran en cuestiones referentes al tiempo consumido por el sistema para el análisis de los documentos o de las consultas, del espacio en disco ocupado por archivos auxiliares, etc. Nuestro estudio se encuentra orientado a aspectos relacionados con la efectividad. La efectividad de un sistema se centra en la adecuación de la información proporcionada por el sistema a las necesidades del usuario.

Los dos índices relacionados con la efectividad que más ampliamente se han utilizado son los denominados *recall* y *precision*.(*) En la definición de ambos índices se diferencia entre documentos relevantes y documentos recuperados. Los documentos relevantes son los idealmente adecuados a la necesidad del usuario, mientras que los recuperados, son los seleccionados por el sistema como adecuados a la consulta. De esta forma, para una consulta se puede hacer una partición del conjunto de documentos como se representa en la figura 2.2 [Salton83a].

(*) Utilizaremos estos términos en Inglés dada la universalidad de su aceptación.



retrel: documentos relevantes recuperados
retnrel: documentos no relevantes recuperados
nretrel: documentos relevantes no recuperados
nretnrel: documentos no relevantes no recuperados

Figura 2.2: Partición de colección de documentos

El recall se define como el cociente entre el número de documentos recuperados por el sistema que son relevantes a la necesidad del usuario, y el número total de documentos que existen en la colección que son relevantes a la necesidad del usuario:

$$recall = \frac{documentos\ relevantes\ recuperados}{documentos\ relevantes\ en\ la\ coleccion}$$

que podemos también escribir, en términos de la figura 2.2:

$$recall = \frac{retrel}{retrel + nretrel}$$

La precision se define como el cociente entre el número de documentos recuperados relevantes a la necesidad del usuario y el número total de documentos recuperados:

$$precision = \frac{documentos\ relevantes\ recuperados}{documentos\ recuperados}$$

en términos de la figura 2.2:

$$precision = \frac{retrel}{retrel + retnrel}$$

Puede verse que, por definición, los valores de los dos parámetros dados un conjunto de documentos y una consulta concreta, se encuentran entre 0 y 1. El valor del recall cuantifica la capacidad del sistema de recuperar el mayor número posible de

los documentos relevantes a la necesidad del usuario. Un recall = 1 supone que el sistema ha recuperado todos los documentos relevantes. Un valor de recall bajo, supone que el sistema ha propuesto al usuario una proporción reducida de los documentos relevantes a su necesidad. El valor de precision cuantifica la capacidad del sistema de no presentar al usuario información irrelevante. Un valor de la precision bajo supone que el sistema ha presentado al usuario un gran número de documentos que no son relevantes a su necesidad. Un valor de precision = 1, supone que todos los documentos propuestos por el sistema se adecuan a la necesidad del usuario.

En sistemas en que se puede hacer variar la cantidad de documentos recuperados para una misma consulta mediante la variación de algún parámetro, se pueden obtener un conjunto de pares de valores precision/recall que nos describen el comportamiento del sistema en función de este parámetro. Habitualmente, se consigue un incremento en uno de los índices a costa de un decremento en el otro.

Comúnmente, el estudio de la efectividad de un sistema se realiza utilizando una colección de documentos determinada y un conjunto de consultas cuyos documentos relevantes han sido previamente asignados [Rijsbergen79, Salton83a, Fagan87, Salton89a, Lewis92]. Un ejemplo típico lo constituye la colección CACM, que contiene 3.204 documentos correspondientes a los abstracts de artículos de la publicación "Communications of the Association for Computing Machinery" entre los años 1958-1979 y 52 consultas con documentos relevantes asignados [Fagan87]. A partir del conjunto de documentos y consultas se obtienen pares de valores precision-recall para cada consulta, calculándose valores medios para el conjunto de consultas. La aplicación de este procedimiento a diferentes sistemas de RI nos permite la comparación de su efectividad en términos de precision y recall. La aplicación del método a un sistema sobre el que realizamos sucesivas modificaciones nos permite estudiar la influencia de éstas en el comportamiento del sistema por lo que respecta a su efectividad.

Los índices de precision y recall han sido los más utilizados para el estudio de la efectividad, pero también han sido los más sometidos a crítica. La suposición que se hace en la definición de ambos parámetros de que se pueda asignar la relevancia o irrelevancia de los documentos a una consulta de una forma "objetiva", exterior al sistema ha sido frecuentemente cuestionada [Rijsbergen89, Salton91b]. Por otra parte, en el cálculo de precision y recall, la relevancia de un documento a una consulta se considera esencialmente un atributo booleano, y no se encuentra especialmente orientado a sistemas en los que se incluye un algoritmo de ordenación por relevancia (ranking) [Raghavan89, Hull93]. También se ha criticado la utilización de recall y precision porque para determinados propósitos es más difícil el estudio mediante dos parámetros simultáneamente que mediante sólo uno, y porque otros parámetros de potencial interés, tales como el tamaño de la colección o el número de ítems recuperados en una búsqueda no se reflejan explícitamente [Salton91b]. En este sentido se han propuesto medidas alternativas tales como el *normalized recall*, *fallout*, *generality*, *sliding ratio*, *expected search length* y *E-measure*, entre otras [Rijsbergen79, Salton83a, Bollman83, Raghavan89, Salton91b, Hull93].

Sin embargo, pese sus posibles deficiencias, recall y precision proporcionan información significativa y de directa interpretación, y constituyen los dos índices más utilizados para el estudio de la efectividad de los sistemas en la actualidad, como resume Hull [Hull93]:

> *"It is common practice to test the performance of new retrieval methods against standard retrieval strategies in a controlled experiment. These experiments are generally based on one or more of the standard IR test collections, and performance (effectiveness) is judged according to the conventional measures of precision and recall. Although this approach to evaluation has often been criticized in recent years, no simple alternatives have yet emerged."*

2.4 El modelo del espacio vectorial

Existe un número importante de modelos orientados a la construcción de sistemas de RI. Los dos más referenciados en la bibliografía son el vectorial [Salton83a] y el probabilístico [Rijsbergen79]. Otros modelos se encuentran orientados a la utilización de diferentes técnicas, como redes neurales [Scholtes93], algoritmos genéticos [Gordon91, Holland92] o redes de inferencia bayesianas [Croft91]. El modelo del espacio vectorial ha constituido la base del mayor número de los experimentos y sistemas desarrollados en el campo [Salton83a, Fagan87, Salton91a, Harman92, Lewis92]. El modelo del espacio vectorial proporciona la base para la implementación de un gran número de operaciones de clasificación de documentos y términos tales como el *agrupamiento de documentos* (*document clustering*), el *agrupamiento de términos* (*term clustering*), la *categorización de textos* (*text categorization*) y el *encaminamiento de texto* (*text routing*) [Lewis92].

Hoy en día, existe una importante evidencia experimental de que el modelo permite el desarrollo de sistemas que presentan una efectividad que es difícil de superar por otras aproximaciones [Fagan87, Salton89a, Salton91a, Lewis92], razón por la cual lo hemos tomado como el modelo en torno al cual desarrollar nuestro estudio.

En el modelo del espacio vectorial [Salton75, Salton83a, Harman92b] se propone la representación de cada documento mediante un vector cuyas componentes son los pesos asociados a los términos utilizados en la representación. Los valores de los pesos pueden ser binarios, enteros, o reales. La dimensión del espacio vectorial es igual al número total de términos utilizado en la representación. De esta forma, si se utiliza para la representación un conjunto de términos $term_i$ con m elementos, cada documento d_j está representado por el vector con m componentes wd_{ji}:

$$(wd_{j1}, wd_{j2}, ..., wd_{jm})$$

La representación de una consulta q_k, se realiza de forma análoga mediante un vector de pesos wq_{ki} asociados a los términos:

	algorithm	architecture	computer	logic	program
Documento1	2.23	5.34	2.45	0.00	0.00
Documento2	3.50	0.00	0.00	3.20	1.51
Documento3	0.00	4.76	3.23	0.00	2.31
Consulta	0.00	0.00	0.00	1.06	0.74

similitudes entre documentos d_i y consulta q:

$$sim(d_1,q) = 0.0000$$
$$sim(d_2,q) = 0.7009$$
$$sim(d_3,q) = 0.2133$$

Figura 2.3: Representación de documentos y consultas, y cálculo de similitud.

$$(wq_{j1}, wq_{j2}, \ldots, wq_{jm})$$

En el modelo se hace la suposición básica de que la distancia relativa entre los vectores de documentos y consultas en el espacio m-dimensional considerado, representa su distancia semántica. De esta forma, se propone la definición de una *función de similitud* (*similarity*) entre un documento d_j y una consulta q_k, basada en el ángulo formado por los dos vectores que las representan. La similitud entre un documento y una consulta viene dada por la expresión del ángulo formado por sus dos vectores asociados:

$$sim(d_j, q_k) = \frac{\sum_{i=1}^{m} wd_{ji} \cdot wq_{ki}}{\sqrt{\sum_{i=1}^{m} wd_{ji}^{\,2} \cdot \sum_{i=1}^{m} wq_{ki}^{\,2}}}$$

Se han propuesto expresiones alternativas a la anterior, habiéndose desarrollado un número importante de experimentos orientados al estudio de su influencia en la el proceso de recuperación, como los realizados en el proyecto SMART [Salton91]. Pero, como concluye Salton:

> *"...this measure is easy to compute and it appears to be as effective in retrieval as other more complicated functions."*

En la figura 2.3 se presenta un ejemplo, en el que el conjunto de términos $term_i$ utilizados para la representación está formado por 5 elementos y los pesos asignados a las componentes de los vectores son reales. Aparece la representación de tres documentos, una consulta, y los valores calculados para sus similitudes.

El valor de la similitud entre un documento y una consulta es la relevancia calculada por el modelo. La asignación de valores de similitud de los documentos a

la consulta permite la ordenación por relevancia con respecto a la necesidad del usuario de los documentos de la colección (ranking) [Harman92].

2.5 Indexación automática de documentos y consultas

En el modelo del espacio vectorial, los documentos y las consultas se representan mediante vectores de pesos. Una serie de técnicas y experimentos se han centrado en la obtención de forma automática de los vectores que representan los documentos tomando como base el análisis de su texto. Parte importante de ellos se desarrollaron a partir en el proyecto SMART anteriormente referenciado.

A partir de la experiencia acumulada, Salton [Salton89a] propone como base para la implementación del proceso de recuperación, el modelo del espacio vectorial en conjunción con la utilización de los siguientes elementos para el proceso de análisis, o indexación:

- pesos de términos
- listas de parada (stoplists)
- extracción de raíces (stemming)
- thesaurus

Estos elementos sirven de base para la obtención automática de los vectores de los documentos a partir de su texto. Permiten además realizar la obtención de los de las consultas expresadas en lenguaje natural, de una forma análoga. Los sistemas desarrollados siguiendo esta aproximación proporcionan una efectividad en el proceso de recuperación que hoy por hoy constituye un estándar difícil de superar.

2.5.1 Pesos de términos

El concepto del *poder de resolución* (*resolving power*) de un término proporciona una base para los métodos de indexación basados en frecuencias de aparición de términos [Rijsbergen79]. El poder de resolución de un término proporciona información acerca de su adecuación como término de indexación.

La definición del poder de resolución se fundamenta en observaciones empíricas relativas a la frecuencia de aparición de las palabras en los textos, tales como la ley de Zipf. La ley de Zipf, establece que, ordenadas las palabras de un texto (o conjunto de textos) por su frecuencia de uso, el producto de su frecuencia de uso por su posición en el ordenamiento es constante:

$$frecuencia \cdot rango \cong cte$$

El poder de resolución se define en función de la frecuencia total de aparición de un término i en la colección de documentos, $totfreq_i$:

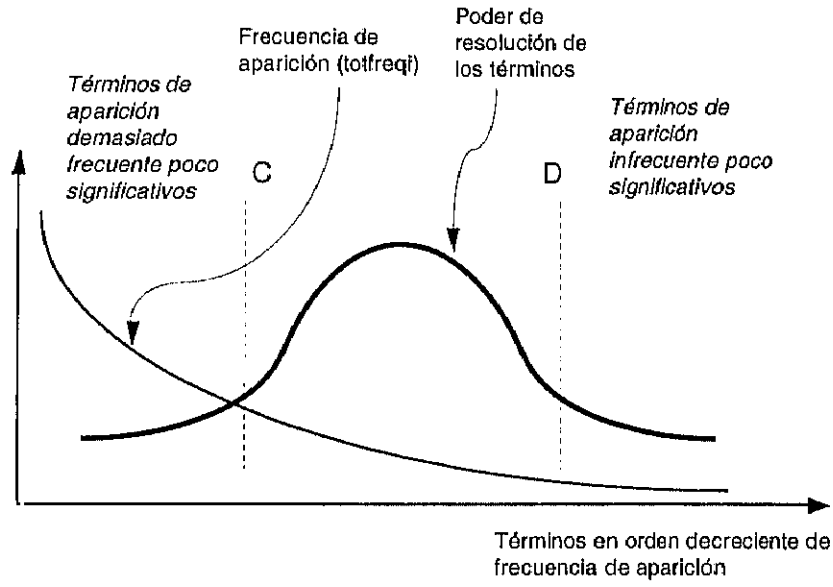


Figura 2.4: Frecuencias de aparición y poder de resolución de los términos.

$$totfreq_i = \sum_{j=1}^{n} tf_{ji}$$

En donde n es el número de documentos de la colección y tf_{ji}, es la frecuencia de aparición (*term frequency*) en el documento j del término i (i.e. el número de veces que aparece el término en el documento). En la definición del poder de resolución, se postula que el poder de resolución de los términos depende de la frecuencia total de aparición de los términos de la forma descrita gráficamente en la figura 2.4. En abcisas se representan los términos en orden decreciente de su frecuencia de aparición en la colección, $totfreq_i$. La curva de trazo fino se corresponde con la frecuencia total de aparición de los términos, $totfreq_i$, obtenida empíricamente y de acuerdo con la ley de Zipf. La curva de trazo grueso se corresponde con el poder de resolución estimado de los términos.

Los términos que se encuentran a la izquierda de C presentan una alta frecuencia de aparición en la colección, tienen poca capacidad de discriminación y son poco representativos como términos de indexación (palabras como "the", "of", "and", "to" podrían ser de este tipo), por lo que no deben utilizarse en la representación. Los términos que se encuentran a la derecha de D son de aparición tan escasa que su presencia o ausencia no afecta de forma significativa al proceso de recuperación. Los términos que tienen mayor poder de resolución se encuentran en la zona media, y son los más adecuados para su utilización como términos de indexación.

El concepto del poder de resolución puede presentarse como el fundamento de los métodos de indexación basados en frecuencias de aparición de los términos [Rijsbergen79], pero tiene una serie de problemas que dificultan su utilización directa para el desarrollo de mecanismos concretos en los sistemas de indexación automática. Uno de ellos es la no definición de un método para la determinación precisa del poder de resolución, ni de los valores de los umbrales C y D. Por otra parte, en estudios teóricos y experimentales posteriores, se han determinado otras medidas de frecuencia de aparición de los términos más efectivas que la frecuencia de aparición absoluta, o total, de los términos en la colección de documentos, $totfreq_i$, en la que se basa.

Una aproximación adecuada al modelo vectorial, apoyada por una importante cantidad de resultados experimentales, se basa en la asignación de pesos a los términos que aparecen en los documentos mediante las expresiones [Salton89a]:

$$wd_{ji} = tf_{ji} \cdot w_i$$

$$w_i = log_2 (n / df_i)$$

Siendo wd_{ji} el valor del peso que se le asigna al término i en el documento j, w_i el peso del término i en la colección, df_i la *frecuencia de documento* (*document frequency*) en que aparece el término i (i.e. el número de documentos en la colección en que aparece el término) y, como en anteriores expresiones, n es el número de documentos de la colección y tf_{ji} es la frecuencia de aparición (term frequency) en el documento j del término i.

Mediante las anteriores expresiones se obtiene una asignación de pesos a los términos en los vectores que representan los documentos, que se encuentra próxima al poder de resolución. Las expresiones asignan valores bajos de w_i a los términos que aparecen frecuentemente en la colección (df_i próximo a n, con $w_i = 0$ para $df_i = n$), con lo que se logra que los pesos de estos términos en los vectores que representan a los documentos, wd_{ji}, tengan siempre valores pequeños. De esta forma se consigue, de una forma análoga al poder de resolución, que los términos que aparecen muy frecuentemente (zona a la izquierda de C en la figura 2.4) tengan escasa importancia en la representación. Por otra parte, el peso que se le asigna a un término en un documento, wd_{ji}, depende de forma directamente proporcional al número de apariciones del término en el documento, tf_{ji}, por lo que a los términos de muy baja frecuencia de aparición (zona a la derecha de D en la figura 2.4), también se les asignan pesos bajos, como en el poder de resolución. Los términos que pueden tener asignados pesos altos en los vectores de los documentos son los que presentan frecuencias de aparición medias, de acuerdo de nuevo con la definición del poder de resolución (zona entre C y D en la figura 2.4).

Por otra parte, también existen diferencias entre la definición del poder de resolución y la de las anteriores expresiones. En concreto, por lo que respecta a la frecuencia de aparición total de los términos, el peso de un término, w_i, depende del número de documentos en que aparece, df_i, a diferencia del poder de resolución, que

depende del número total de veces en que aparece en el conjunto de los documentos, $tot freq_i$. Esta diferencia, como ya hemos señalado anteriormente, introduce mejoras en el proceso de recuperación de acuerdo con estudios teóricos y experimentales [Salton83a, Salton89a].

2.5.2 Listas de parada

Las *listas de parada* (*stoplists*) [Fox92a] se utilizan en el análisis de los documentos para la eliminación de una serie de palabras que no resultan útiles para la obtención de términos de indexación. De acuerdo con lo visto en puntos anteriores, términos obtenidos a partir de estas palabras tienen poderes de resolución, o pesos, nulos. Ejemplos de estas palabras en Inglés son "the", "a", "to", etc. Estas listas se obtienen en estudios orientados específicamente a ello, a partir de un corpus de textos lo suficientemente representativo del idioma considerado. Por ejemplo, puede encontrarse en [Rijsbergen79] una lista de parada de 250 términos y en [Fox92] una de 425 obtenida a partir del *Brown Corpus* [Edwards92]. Como dato concreto, las diez palabras que aparecen más frecuentemente en Inglés pueden constituir entre el 20 y el 30 por ciento de los "tokens" de un documento [Fox92a].

La política de stoplists depende del dominio de la base de datos concreta. Por ejemplo, el ORBIT Search Service, de propósito general, trabaja con una stoplist de sólo ocho palabras: "and", "an", "by", "from", "of', "the" y "with" [Fox92a]. Otros sistemas utilizan stoplists de mayor número de términos, como la de 425 términos anteriormente citada, que es utilizada en el sistema WAIS [TMC91].

2.5.3 Extracción de raíces

Los *algoritmos de extracción de raíces* (*stemming*), o de eliminación de sufijos, se encuentran orientados a obtener un único término a partir de diferentes palabras que constituyen esencialmente variaciones morfológicas con un mismo significado [Frakes92b, Krovetz93]. Como ejemplo podemos considerar la obtención del término ANALY a partir de "analysis", "analyzer" y "analyzing".

El resultado del algoritmo debe ser una misma forma canónica para las diferentes variantes morfológicas de una palabra, que no tiene por qué ser, necesariamente, la raíz lingüística. En la bibliografía se pueden encontrar diferentes tipos de algoritmos de stemming [Frakes92b]. Se pueden introducir dos tipos de errores en este tipo de algoritmos. El primero de ellos es el error de *infrarradicación* o *understemming*, que resulta de obtener diferentes formas canónicas para palabras que deberían proporcionar una misma por tener un mismo significado; por ejemplo para "hope" y "hopping" obtener HOP Y HOPP, respectivamente. El segundo error es el de *sobrerradicación* u *overstemming*, que es su complementario: obtener la misma forma canónica para palabras que deberían tenerlas distintas, por diferir no en variaciones morfológicas, sino en su significado; por ejemplo, para "capital", "capitulate" y "capitol" obtener CAPIT.

La elección de un algoritmo concreto depende de la política seguida en la construcción de un sistema concreto. Para evitar errores en el proceso de recuperación interesan algoritmos que minimicen la sobrerradicación e infrarradicación. Algoritmos menos sofisticados que tiendan al overstemming y eviten el understemming (se disminuye el número de índices) pueden adecuarse a sistemas en que aspectos de eficiencia en tiempo y espacio juegan un papel prioritario, pero conllevan una disminución en la efectividad del proceso [Frakes92b].

2.5.4 Frases de términos

Las *frases de términos* (*term phrase*) [Salton83a, Lewis92] se orientan a la obtención de términos de indexación con un significado más preciso que el de los términos obtenidos directamente a partir de las individualmente.

Una frase de términos es una tupla de términos y constituye en sí misma un nuevo término de indexación. Si consideramos el caso de frases de términos formadas por parejas, una frase $phrase_{ik}$ se forma a partir de los términos $term_i$ y $term_k$. Es de esperar que la frecuencia de aparición de la frase $phrase_{ik}$ (que a partir de ahora se considera término de indexación) en la colección de documentos sea menor que la de los términos $term_i$ y $term_k$, y por lo tanto tenga una interpretación más especifica que la de sus componentes individuales, y por tanto, potencialmente, un mayor poder de resolución. Por ejemplo, a partir de dos términos generales como "program" y "computer" se obtiene la frase de términos, más específica "computer program".

Existen diversos métodos de generación de frases de términos [Salton83a, Croft91, Lewis92], incluidos los basados en el análisis sintáctico [Fagan87] (esta última aproximación la trataremos en capítulos siguientes). Una aproximación para la obtención de frases de términos se basa en el *valor de cohesión de una pareja de términos*, que puede definirse en función de la frecuencia de aparición de los términos [Salton83a]:

$$cohesion_{kh} = pairfreq_{kh} / (totfreq_k \cdot totfreq_h)$$

Siendo $pairfreq_{kh}$ la frecuencia de aparición de los dos términos simultáneamente y $totfreq_i$ la frecuencia aparición total de aparición del término $term_i$. Para utilizar la anterior fórmula es necesario definir el contexto en el que se considera que dos términos co-aparecen. Los mejores resultados se obtienen cuando consideramos que dos términos co-aparecen cuando lo hacen en la misma oración, o con un número máximo de palabras entre ambos de un orden parecido. Se seleccionan como términos las parejas que tengan un valor de cohesión por encima de un determinado umbral.

La utilización de frases de términos, a diferencia de los otros elementos tratados en este punto, no se incluye en la aproximación de Salton [Salton89a] debido a que los resultados obtenidos en experimentos destinados a evaluar su utilización no han

413	LONGITUDINAL	415	ANTENNA	(ffa, flow, blood, serum, secretium)
	TRANSVERSE		KLYSTRON	
414	CRYOGENIC		RECEIVER	(change, increase, effect, response, pattern)
	CRYOTRON		TRANSMITTER	(day, hour, week, year, month, hr, time)
	PERSISTENT-CURRENT		WAVEGUIDE	(rat, mouse, animal, dog, female, infant)
	SUPERCONDUCT			

(a) Thesaurus (b) Lista de asociación

Figura 2.5: Fragmentos de thesaurus y de lista de asociación de términos.

proporcionado resultados definitivos. Por otra parte, las frases de términos juegan un papel importante en sistemas que trataremos en capítulos siguientes.

2.5.5 Thesaurus

Un thesaurus proporciona una agrupación o clasificación de términos en un determinado dominio o área en categorías denominadas *clases*. En la figura 2.5.a aparece un ejemplo de thesaurus [Salton83a]. Pueden utilizarse diversos tipos de thesaurus en la indexación automática o manual de los documentos [Srinivasdan92]. En una aproximación basada en el análisis automático de textos y consultas, la utilización del thesaurus permite identificar términos lexicográficamente diferentes como equivalentes semánticamente (i.e. sinónimos). La utilización de thesaurus se puede realizar durante el proceso de indexación de los documentos (utilizando como términos de indexación las clases del thesaurus), o durante el análisis de las consultas (realizándose una expansión de los términos de la consulta mediante sus equivalentes) [Srinivasdan92, Qiu93].

Con vistas a cuestiones que tratamos en capítulos siguientes, nos interesa considerar las diferenciaciones:

- Thesaurus generales / específicos del dominio
- Thesaurus construidos manualmente / automáticamente

Thesaurus de propósito general y específicos del dominio

Desde el punto de vista de su utilización en el proceso de recuperación, considerar dos palabras equivalentes depende del dominio (o conjunto de documentos) sobre el que el sistema trabaja. Por ejemplo, sobre un dominio general, poco específico (e.g. conjunto de noticias de actualidad diversas relativas a política, economía, ciencia, etc.), términos como "cryogenic" y "superconduct" se pueden considerar equivalentes a efectos de indexación, como en la figura 2.5.a. Esto, sin embargo, no es válido sobre un conjunto de documentos formado exclusivamente por artículos de microelectrónica. Criterios basados en la frecuencia de aparición de los términos pueden contribuir a decidir cuándo incluir los términos en clases más generales o no [Salton83a, Srinivasdan92]. En cualquier caso, disponer de un thesaurus específico del dominio contribuye a la mejora de la efectividad del proceso de recuperación.

Construcción automática de thesaurus

La construcción manual de un thesaurus es un proceso costoso. Existen métodos de construcción de thesauri semiautomáticos y completamente automáticos [Strinivasan92]. Los métodos de construcción de *listas de asociación de términos*, equivalentes a las clases del thesaurus anteriormente comentadas, permiten su obtención a partir del análisis de los documentos. Las listas de asociación son conjuntos de palabras con significado parecido a efectos de recuperación. La suposición fundamental en que se basa la construcción de estas listas es que [Grefenstette92]:

"similar terms appear in similar contexts"

La siguiente expresión se puede utilizar para el cálculo de la similitud entre dos términos $term_j$ y $term_k$ en una colección de n documentos [Salton 83, Strinivasan92]:

$$sim(term_j, term_k) = \sum_{i=1}^{n} wd_{ij} \cdot wd_{ik}$$

En donde los valores wd_{ij} y wd_{ik} son los pesos de los términos $term_j$ y $term_k$ en el documento i.

De esta forma, términos que aparecen simultáneamente en un número importante de documentos resultan con valores de similitud altos, mientras que términos que no aparecen en los mismo contextos, resultan con valores de similitud bajos, o nulos. A modo de ejemplo, si las palabras "permission" y "authorization" aparecen a la vez en un gran número de documentos, obtendremos una similitud para ellas importante. La formación de las asociaciones, o grupos, de términos se realiza a partir de sus valores de similitud pudiéndose utilizar diferentes algoritmos de agrupación (clustering) [Salton83a, Strinivasan92]. En la figura 2.5.b aparece un ejemplo de listas de asociación obtenidas a partir de un conjunto de "abstracts" médicos [Greffenstette92].

2.6 Resumen y conclusiones

En este capítulo hemos hecho una presentación del proceso de Recuperación de la Información adaptada al enfoque de nuestro estudio, revisando los aspectos que tienen una mayor relevancia para este trabajo. Nuestro enfoque se ha centrado en métodos de recuperación orientados al desarrollo de sistemas que tienen como entradas de datos fundamentales documentos y consultas en lenguaje natural.

Se han presentado los aspectos centrados en la efectividad como los más relevantes a nuestro trabajo. Hemos tratado la utilización de los índices de recall y precision, sobre un conjunto de documentos y consultas con criterios de relevancia previamente asignados, como los más adecuados para el estudio de la efectividad de los sistemas.

La utilización de una serie de elementos introducidos en este capítulo proporcionan la base para la implementación del proceso de recuperación de forma tal que, hoy por hoy, la efectividad conseguida es difícil de superar por otras aproximaciones. Estos son, en concreto: el modelo del espacio vectorial, la utilización de pesos de términos basados en su frecuencia de aparición, uso de listas de parada y algoritmos de extracción de raíces. También se han tratado otros elementos que pueden contribuir en determinadas ocasiones a la mejora de la efectividad y que son de relevancia para capítulos siguientes, tales como las frases de términos y los thesauri.

CAPITULO 3

LA RECUPERACION DE INFORMACION EN BIBLIOTECAS DE COMPONENTES SOFTWARE: EL SISTEMA ARGOS

3.1 Introducción

Un número importante de entornos de desarrollo de sistemas informáticos proporcionan al programador un conjunto de componentes software reutilizables. Estas colecciones o bibliotecas de componentes software incluyen habitualmente documentación textual en formato electrónico relativa a su funcionalidad y forma de utilización. El sistema Argos ha sido concebido como un sistema de ayuda a la utilización de estas bibliotecas de componentes. Argos utiliza como principal fuente de información la documentación existente en lenguaje natural en el propio entorno. Este sistema se ha desarrollado para un dominio concreto: el sistema operativo UNIX, y procesa la documentación correspondiente al manual de referencia existente en el mismo entorno (figura 3.1). Por otra parte, los principios utilizados en Argos han sido adaptados al desarrollo de sistemas similares en otros dominios.

La concepción del sistema Argos está orientada a la definición del marco de aplicación concreto sobre el que poder investigar la utilización de diferentes tipos de técnicas relacionadas con la Recuperación de Información. De esta forma, el sistema se ha desarrollado con vistas a lograr una utilidad práctica inmediata: se han tenido presentes criterios tales como los relacionados con la interacción hombre-máquina y la facilidad de utilización por el usuario. Por otra parte, el sistema Argos nos ha servido en nuestro trabajo para definir el marco concreto en el que integrar técnicas de Procesamiento de Lenguaje Natural en la Recuperación de Información. El procesamiento de la documentación existente en el entorno, así como la formulación por parte del usuario de sus necesidades en Lenguaje Natural juegan un papel central en el sistema.

En este capítulo presentamos el sistema Argos en su conjunto y la forma en que se han implementado las funciones de procesamiento de textos y consultas, y el cálculo de su similitud. En Argos se han utilizado técnicas basadas fundamentalmente en

```
DF(1V)                    USER COMMANDS                    DF(1V)
NAME
   df - report free disk space on file systems
SYNOPSIS
   df [ -a ] [ -i ] [ -t type ] [ filesystem... ]
   [ filename... ]
SYSTEM V SYNOPSIS
   /usr/5bin/df [ -t ] [ filesystem... ] [ filename... ]
AVAILABILITY
   The System V version of this command is available with the System V
   software installation option. Refer to Installing SunOS 4.1 for information
   on how to install optional software.
DESCRIPTION
   df displays the amount of disk space occupied by currently mounted file
   systems, the amount of used and available space, and how much of the file
   system's total capacity has been used. Used without arguments, df
   reports...
```

Figura 3.1: Fragmento de documentación de la orden *df* del manual de UNIX.

criterios estadísticos. En concreto, el modelo del espacio vectorial, la utilización de pesos de términos, extracción de raíces y listas de parada.

En los puntos siguientes tratamos en primer lugar las formas en que los métodos de recuperación de información pueden ser utilizados en bibliotecas de componentes software. A continuación presentamos los objetivos y la estructura del sistema Argos. En un punto siguiente, la forma en que se ha implementado el subsistema que implementa el procesamiento de los textos y las consultas, y el cálculo de su similitud. Finalmente, se presenta un resumen y conclusiones.

3.2 Recuperación de información y bibliotecas de componentes software

La *reutilización del software* es un aspecto importante del proceso de Ingeniería del Software que puede proporcionar incrementos importantes en la calidad y la productividad [Biggerstaff89b, Rumbaugh91, Booch91, Krueger92, Pressman93, Mili95]. Con vistas a promover la reutilización, una cantidad importante de bibliotecas de componentes software han aparecido durante los últimos años, teniendo todas ellas un aspecto común: sus grandes dimensiones [Biggerstaff89, Mili95]. Paralelamente, un número creciente de trabajos se ha orientado al estudio de la aplicación de técnicas de Recuperación de la Información para facilitar determinados aspectos de la reutilización del software en estos entornos [Frakes89, Frakes94, Mili95].

Mediante el término general *componente software* (*software component*) nos referiremos en las siguientes páginas a cualquier módulo reutilizable que consideremos, tanto a nivel de código como de diseño. Ejemplos concretos de componentes software que aparecen en la literatura son una función en C, un

método o una clase de Smalltalk, un paquete de Ada, o una orden de UNIX. Utilizaremos indistintamente los términos *colección de componentes* o *biblioteca de componentes*.

Una serie de problemas que presentan las grandes colecciones de componentes software son semejantes a los ya identificados en el campo de la RI en torno a las bibliotecas de documentos. Los trabajos de RI orientados a facilitar la reutilización de bibliotecas de componentes se centran en problemas relativos a su clasificación, almacenamiento y recuperación [Frakes89, Frakes94]. El paralelismo existente entre estos problemas que plantean las bibliotecas de componentes software y las documentales, nos permite utilizar el término *recuperación de componentes* para referirnos al proceso a realizar en bibliotecas de componentes software, análogo al de recuperación de información, en bibliotecas documentales.

Para el propósito de nuestro estudio, se pueden diferenciar dos aproximaciones en la utilización de técnicas de RI en determinados aspectos de la gestión de bibliotecas de componentes software [Fernández-Chamizo95, Mili95]: la basada en el conocimiento (o en la indexación manual) y la basada en la indexación automática. En la basada en la indexación automática, la fuente de información que permite la indexación de los componentes es la documentación en lenguaje natural existente en el entorno. Los sistemas que hemos desarrollado y presentamos en nuestra tesis siguen la aproximación basada en la indexación automática, pero, como veremos en este y en capítulos siguientes, utilizan también algunos elementos tomados de los sistemas basados en la indexación manual.

3.2.1 Recuperación de componentes basada en la indexación manual

En los sistemas basados en la indexación manual, se proporciona a los usuarios un mecanismo para representar (o indexar) las componentes y seleccionar las más relevantes a sus necesidades. En estos sistemas [Prieto-Díaz87, Sommerville88, Devambu91, Prieto-Díaz91, Katalagarianos95], cada componente se representa mediante un registro, o mediante una estructura de tipo *frame* en sistemas basados en el conocimiento [Frost86]. Esta información debe ser introducida manualmente en el sistema por el usuario (o bibliotecario) para cada componente que se incluya en la colección, con lo que se genera una base de datos (o base de conocimiento) con información relativa a las funcionalidades de las componentes. La formulación de las consultas por parte de los usuarios se realiza mediante el relleno parcial de plantillas, correspondientes a un registro o frame, que representa su necesidad y que se utiliza para hacer "matching" con los existentes en el sistema correspondientes a todos los componentes de la biblioteca.

En la literatura encontramos dos sistemas que son especialmente relevantes para nuestro estudio. En el primero de ellos [Prieto-Díaz87], se propone un esquema de clasificación basado en facetas. El esquema de facetas se diseñó para la caracterización de fragmentos de software (componentes) de tamaño medio (50 a 200 líneas de código). El conjunto de facetas es {Función, Objetos, Medio, Tipo de sistema, Area funcional, Localización}. En la definición del esquema de facetas se

incluyen los nombres concretos de éstas y los posibles valores que pueden ser asignados a cada una de ellas. En la tabla siguiente se muestra parte de los valores posibles para cada una de las facetas:

Function	Objects	Medium	SystType	FuncArea	Setting
add	arrays	array	assembler	accounts	advertising
append	blanks	buffer	compiler	auditing	association
close	buffers	disk	DB mang.	billing	barber_shop
...					

El esquema de facetas es utilizado por el bibliotecario encargado del mantenimiento. Este completa manualmente un registro por cada componente, dando a cada una de las facetas los valores que mejor lo describan. Un ejemplo sería [Prieto-Díaz87]:

```
description( function(compress),
             objects(files),
             medium(disk),
             syst_type(file_handler),
             funct_area(db_manager),
             setting(catalog_sales) ).
```

En un trabajo posterior [Prieto-Díaz91] se presenta un sistema con la misma organización para la gestión de órdenes de UNIX, basado en el conjunto de facetas {Acción, Objeto, Estructura de Datos, Sistema}.

El segundo sistema relevante a nuestro trabajo es el de Wood y Sommerville [Wood88]. Se propone un sistema para la gestión de dos tipos de bibliotecas de componentes. La primera de ellas se encuentra formada por paquetes de Ada (packages) en un entorno de desarrollo de aplicaciones relacionadas con aeronáutica, y la segunda por órdenes del S.O. UNIX. Para los comandos de UNIX utilizan una representación de tipo frame con slots basados en las funciones gramaticales utilizadas por la teoría de la dependencia conceptual (actor, object, destination, etc.) Por ejemplo, la representación de la orden *more*, que presenta paginado un archivo en la terminal del usuario, es:

```
frame(   action(print),
         actor(more),
         object(file),
         destination(terminal) ).
```

Los propios autores señalan la adecuación de esta representación basada en ideas originalmente introducidas en PLN, para las descripciones de las órdenes de UNIX y las solicitudes de los usuarios:

> *"Our system is based on ideas derived from natural language processing where an attempt is made to understand the semantics of a component description or retrieval request"*

3.2.2 Recuperación de componentes basada en la indexación automática

En esta aproximación la documentación en lenguaje natural debe formar parte de la propia biblioteca de componentes. El proceso de indexación de las componentes se realiza de forma automática a partir del análisis de los textos de la documentación asociada. En esta aproximación es necesario que tanto la biblioteca de componentes como la documentación existente reúnan una serie de requisitos, en [Maarek91] se señalan los siguientes como principales:

- *Existencia de documentación.* Debe encontrarse disponible la documentación en Lenguaje Natural relativa a la funcionalidad y utilización de los componentes.

- *Relación uno-a-uno entre documentos y componentes.* Con vistas a utilizar un documento como una descripción de un componente software, debe existir una relación uno-a-uno entre el conjunto de componentes y la documentación disponible. Idealmente, esta relación uno-a-uno debe existir previamente o ser fácilmente inferible. En algunos casos puede resultar necesario cierto trabajo manual, como por ejemplo, en bibliotecas de clases, en las que no todas las clases se encuentran documentadas individualmente [Helm91]. En cualquier caso este trabajo suele conllevar un esfuerzo menor que el de un análisis del dominio completo [Prieto-Díaz90], si no es así, la aproximación basada en la documentación puede resultar no deseable.

- *Nivel de abstracción de los documentos.* Resulta deseable que los documentos considerados se encuentren a un mismo nivel de abstracción. Por ejemplo, intercalar documentos que describen detalles de implementación con documentos que describen funcionalidades de alto nivel puede conllevar decrementos en la efectividad, con bajos valores en los índices de recall y precision.

Una colección de componentes reutilizable que satisface estos requisitos es el conjunto de herramientas que proporciona el sistema operativo UNIX como órdenes accesibles a los usuarios, junto con la documentación que proporciona el entorno en su manual. Existen otras bibliotecas de componentes comerciales, como AIX, ULTRIX, X, Motif, VisualWorks y OpenLook, por citar algunos ejemplos, que utilizan el mismo modelo de documentación que resulta especialmente adecuado para esta aproximación.

Un sistema que sigue esta aproximación es Guru [Helm91, Maarek91], que emplea técnicas basadas en frecuencias aparición de términos para el procesamiento de documentos y consultas de los usuarios análogas a las tratadas en el capítulo anterior. Así mismo, el sistema Proteus [Frakes94], utiliza también la documentación existente en la biblioteca de componentes, y se basa en la utilización de operadores booleanos para la formulación de consultas. En este último sistema no se utilizan criterios basados en la frecuencia de aparición de los términos.

3.3 El sistema Argos

El sistema Argos es un sistema de ayuda a la utilización del conjunto de órdenes de UNIX en el que la utilización de la documentación existente en el entorno mediante técnicas de indexación automática juega un papel fundamental. Tratamos a continuación los objetivos marcados en el desarrollo del sistema, su funcionalidad y su arquitectura.

3.3.1 Objetivos del sistema

En la concepción del sistema Argos se han tenido como objetivos principales los tres siguientes:

1. *Definición de un sistema práctico*. Para la consecución de un sistema efectivo, orientado a su utilización práctica por un conjunto de usuarios significativo, ha sido necesario la definición de una serie de funcionalidades y la utilización de una serie de elementos orientados a facilitar el proceso de recuperación de información adicionales a las técnicas tratadas en el capítulo 2. Por ejemplo, se hace uso de criterios de *interacción hombre-máquina* [Maddix90], ideas de *hipertexto* [Smith88, Balasubramanian93], así como de elementos relacionados con *modelado de usuario* [Kobsa89, Kobsa94].

2. *Definición de una aproximación generalizable*. El sistema Argos se ha desarrollado para el dominio específico del conjunto de utilidades que proporciona el sistema operativo como órdenes y que se encuentran documentadas en la sección 1 del manual [Sun89]. Sin embargo, la aproximación seguida en Argos resulta de fácil adaptación a otras bibliotecas de componentes software, especialmente a aquellas que cumplen los requisitos establecidos anteriormente para aproximación a la recuperación de componentes basada en la indexación automática. En concreto, adaptaciones de la aproximación seguida en Argos se han realizado a la biblioteca de clases de Smalltalk del entorno VisualWorks[González95].

3. *Definición de un marco de aplicación concreto orientado a la investigación*. Argos ha sido concebido como un marco de aplicación concreto, sobre un dominio concreto (i.e. la documentación de UNIX) orientado al estudio y evaluación de la integración de diferentes tipos de técnicas en el proceso de recuperación de información [Buenaga93a, Buenaga93b, Fernandez-Manjón93]. Argos, de hecho, proporciona la definición del marco concreto de aplicación de los sistemas Ares [Buenaga94, Buenaga95] y Aran [Fernandez-Manjón95a, Fernandez-Manjón95b]. Estos sistemas se centran en el uso de técnicas de procesamiento de lenguaje natural, y en la inclusión de una representación explícita del dominio, respectivamente.

Estos tres aspectos del sistema son tratados en párrafos siguientes y pueden encontrarse detalles adicionales en la bibliografía [Buenaga93a, Buenaga93b, Fernández-Manjón93, Arroyo93, Fernández-Manjón94, Buenaga94, Fernández-Manjón95a, Fernández-Manjón95b, Díaz95, Buenaga95]. El punto 3 es el más

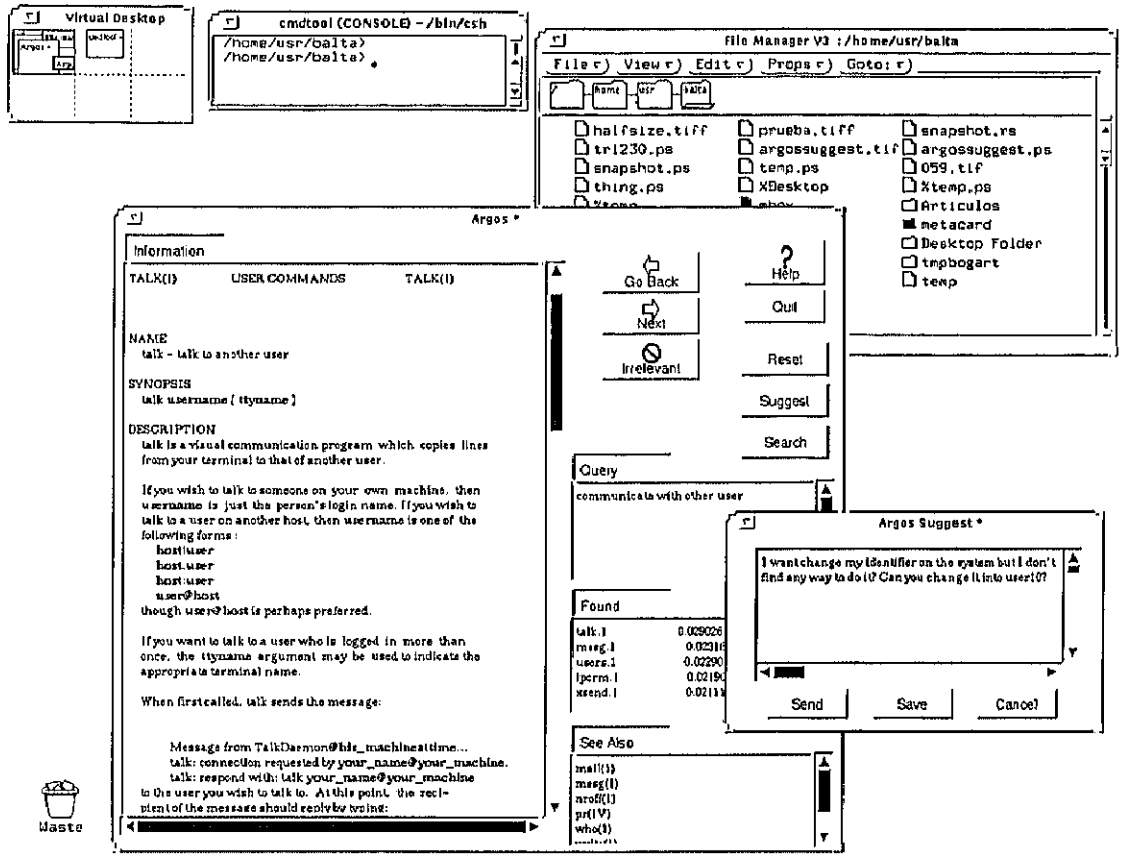


Figura 3.2: Interfaz del sistema Argos en el entorno X Window.

importante para nuestro trabajo y de él trataremos con más detalle en este capítulo y en siguientes.

3.3.2 Funcionalidades del sistema

En la versión actual de Argos el usuario se encuentra trabajando en el entorno X Window [Johnson89] en una ventana de órdenes. El sistema es accesible como una aplicación más. En la figura 3.2 podemos ver el interfaz del sistema. Las principales funcionalidades implementadas por el sistema y los criterios utilizados para su definición son:

- *Especificación de consultas en lenguaje natural.* El usuario utiliza el lenguaje natural (Inglés) se utiliza como medio para especificar su necesidad. De esta forma, cuando el usuario se encuentra ante una determinada necesidad, puede formular en el panel *query* consultas como "communicate with other user", "copy a file to a remote machine", "get the name of the current directory", etc. Conviene subrayar que el lenguaje natural se utiliza sólo para expresar la necesidad del usuario de forma declarativa. Las órdenes en sí

mismas (e.g. la orden de comenzar la búsqueda, la de presentar un documento o la de finalizar) son realizadas mediante la interacción basada en otros tipos de lenguajes (e.g. botones y menús).

- *Interacción basada en un lenguaje de manipulación directa.* Bajo el término *lenguaje de manipulación directa* (*direct manipulation language*) queda englobada la interacción basada en la utilización uso de botones, menús desplegables, ventanas y ratón como dispositivo señalizador. La asignación de los lenguajes a los diferentes elementos de interacción, buscando su mejor adecuación, se ha realizado siguiendo criterios de *Interacción Hombre Máquina* (*Human Computer Interaction, HCI*) [Maddix90]. De esta forma, la solicitud de ejecución de órdenes concretas se realiza mediante botones (e.g. la orden de comenzar la búsqueda mediante *search* y la de fin de ejecución mediante *quit*). Utilizar el lenguaje natural para este tipo de funciones resulta factible, pero probablemente inadecuado [Bates87, Maddix90].

- *Presentación de la documentación en lenguaje natural.* La selección de las órdenes relevantes a la necesidad del usuario se basa en el cálculo de la similitud entre la necesidad del usuario y los documentos asociados a las órdenes. En la ventana *found* se presentan los nombres de las órdenes con sus valores de relevancia. En la ventana *information* el texto de la orden más relevante, o de las seleccionadas por el usuario mediante el ratón.

- *Funciones de navegación basadas en hipertexto.* El proceso formado por la secuencia de consultas y la presentación de la información por parte del sistema, puede concebirse como uno de navegación por un *hipertexto* (*hypertext*) [Conklin87, Smith88, Nielsen90, Cybulski92, Allan95], generado automáticamente. En este hipertexto los *nodos* son los ítems de información del manual (i.e. cada documento asociado a cada orden) y los *enlaces* quedan definidos por su secuencia de presentación. Con este fin se incluyen funciones de navegación especialmente adecuadas para este tipo de sistemas, como *go back* y *next*. El cálculo de la similitud de los documentos con las consultas constituye la base de la generación de enlaces dinámicos. La información sobre órdenes relacionadas incluidas en la sección *see also* del final de cada documento proporciona enlaces estáticos adicionales.

- *Realimentación.* En Argos se incluyen funciones orientadas a la introducción de *realimentación* (*feedback*) en el proceso de recuperación [Salton89a, Aalbersberg92, Harman92a, Harman92c]. La realimentación se basa en la formulación de consultas mediante la utilización de documentos proporcionados por el sistema al usuario y proporciona importantes mejoras en la efectividad del proceso de recuperación en su conjunto. En concreto, si parte del texto que aparece en *information* es especialmente relevante, el usuario puede seleccionarlo mediante el ratón, utilizándose este fragmento del texto como una nueva consulta. Por otra parte, cuando la documentación que se presenta en *information* es irrelevante al problema del usuario, éste puede indicarlo mediante el botón *irrelevant*. Esta información relativa a la irrelevancia es utilizada en el procesamiento de siguientes consultas relativas

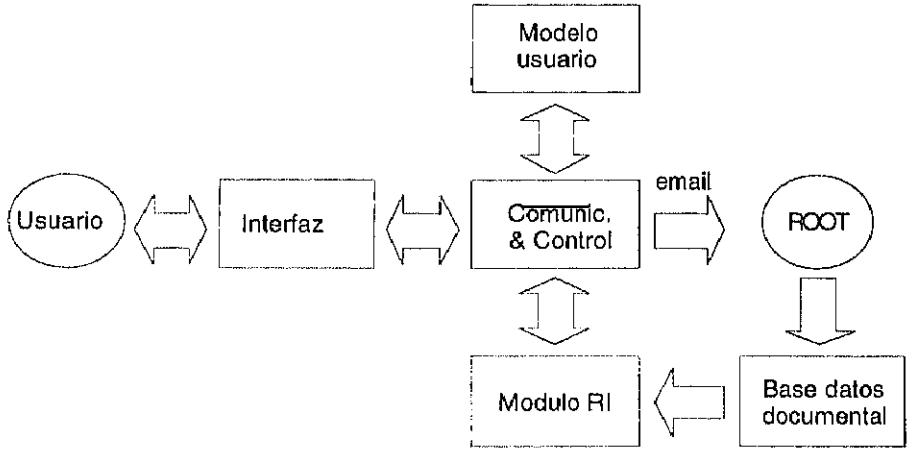


Figura 3.3: Estructura de módulos del sistema Argos

a una misma necesidad o tema de búsqueda. El botón *reset* sirve para indicar un cambio en la necesidad o tema de búsqueda.

- *Adición de información textual.* Además los usuarios pueden incluir información específica, adicional a la existente en el manual de UNIX. Información que puede resultar de mayor interés para el grupo de trabajo del usuario y más adecuada a su entorno de trabajo, o simplemente, más concisa. Tras la edición de esta información mediante la orden *suggest* (figura 3.2), los documentos son enviados por Argos, mediante correo electrónico, al administrador del sistema para su supervisión. De esta forma, la base de datos documental utilizada por Argos está formada por los ficheros del manual y las sugerencias redactadas por los usuarios, obteniéndose de este modo capacidades propias de los *entornos cooperativos supervisados* [Fernández-Manjón93].

3.3.3 Estructura del sistema

Argos consta de cuatro módulos (figura 3.3): *interfaz*, que se ocupa de la interacción con el usuario; *modelo de usuario*, que contiene la información disponible sobre el usuario; *recuperación de información*, que implementa las funciones de indexación de los documentos y cálculo de la similitud; y un módulo de *comunicación y control*, que proporciona la adecuada integración del resto de los módulos. La base de datos documental está formada por los archivos de texto del manual de UNIX y los documentos añadidos por los usuarios y supervisados por el superusuario (root). En su diseño se ha tratado en todo momento de mantener una estructura modular de forma que se pueda modificar o sustituir alguna de sus partes con facilidad.

- *Modelo del usuario.* El módulo del modelo de usuario (MU) es un módulo cuyo objetivo es adecuar la información que se presenta a cada usuario concreto. Para ello utilizamos un *modelo de usuario incremental* [Kobsa92]. La información utilizada en el MU proviene tanto de la interacción con el usuario

(a través de sus criterios de relevancia o feedback), como del tipo de usuario en que se clasifica. Se ha hecho una clasificación en categorías de las órdenes de UNIX, basándose en su posición normal en la curva de aprendizaje. Las categorías identificadas son: básica, intermedia, avanzada, supervisor, programador y específica. Al usuario se le asigna un determinado nivel de conocimiento (e.g. en función del estudio del histórico de las órdenes que ha utilizado en el sistema operativo) y esto se utiliza para modificar las búsquedas y el orden de presentación [Fernández-Manjón93, Fernández-Manjón95b]. Por ejemplo, partes del manual de UNIX tratan de utilidades para la programación en lenguaje C. Aunque estos documentos pueden presentar una cierta similitud con determinadas consultas, el MU estima que sólo le interesan a los usuarios que realizan labores de programación.

- *Interfaz*. En el diseño del interfaz se ha tenido en cuenta la sencillez de manejo y la uniformidad. Es importante evitar la sobrecarga cognoscitiva del usuario y la dificultad de uso del sistema [Downton91, Fernández-Manjón94]. En el diseño del interfaz de Argos se han tenido presentes criterios de interacción hombre-máquina y elementos tomados del diseño de interfaces a sistemas de hipertexto mencionados en párrafos anteriores. Detalles adicionales de implementación se pueden encontrar en [Arroyo93].

- *El módulo de recuperación de información*. El módulo de recuperación de información (RI) realiza las órdenes de búsqueda que recibe del módulo de control del sistema. El contenido esencial de las órdenes de búsqueda que recibe el módulo de RI es una lista de palabras, confeccionada a partir de la consulta del usuario y la información existente en el modelo del usuario. Esta lista de palabras caracteriza la necesidad del usuario y es la utilizada para realizar el cálculo de la similitud de los documentos en la base de datos con respecto a la necesidad del usuario.

En el siguiente apartado proporcionamos más detalles acerca de la implementación del módulo de recuperación de información, el más directamente relacionado con el presente trabajo.

3.4 El subsistema de recuperación de información

Las funcionalidades del módulo de recuperación de información han sido encapsuladas en un subsistema, Superman, que es utilizado desde Argos y que también puede ser utilizado desde otras aplicaciones, incluido el intérprete de órdenes del sistema operativo. El sistema implementa, esencialmente, un algoritmo de cálculo de similitud entre consultas y documentos. Se basa en el modelo del espacio vectorial y la utilización de pesos de términos, extracción de raíces y listas de parada. El sistema se ha desarrollado para el procesamiento de la sección 1 del manual de referencia, pero resulta de fácil adaptación a otras colecciones de descripciones de componentes diferentes.

a	and	away	best	come
about	another	b	better	could
above	any	back	between	d
across	anybody	backed	big	did
after	anyone	backing	both	differ
again	anything	backs	but	different
against	anywhere	be	by	differently
all	are	became	c	do
almost	area	because	came	does
alone	areas	become	can	done
along	around	becomes	cannot	down
already	as	been	case	down
also	ask	before	cases	downed
although	asked	began	certain	downing
always	asking	behind	certainly	downs
among	asks	being	clear	during
an	at	beings	clearly	

Figura 3.4: Subconjunto de la lista de parada utilizada en Superman.

3.4.1 Método de indexación y cálculo de similitud

En el desarrollo del sistema Superman se ha optado por la utilización de un conjunto de técnicas que aúnan dos ventajas frente a otras aproximaciones: su sencillez de implementación y su efectividad [Salton89a, Frakes92a]. La aproximación seguida se basa en las técnicas de indexación descritas en el capítulo 2. A continuación se detallan los elementos utilizados para el desarrollo del sistema.

1. *Uso de lista de parada* (*stoplist*) para la eliminación de palabras de alta frecuencia de aparición. Se utiliza una lista compuesta por 429 palabras [Fox92a]. En la figura 3.4 se incluyen las 84 primeras palabras de la lista utilizada. El algoritmo de eliminación de estas palabras se basa en el uso de un autómata de estados finitos [Fox92a].

2. *Uso de una rutina de eliminación de sufijos* (*stemming*). Para la implementación del procedimiento de extracción de raíces nos basamos en el algoritmo de Porter, que utiliza un autómata de estados finitos e incluye 88 reglas de eliminación de sufijos [Frakes92b]. Las reglas se encuentran agrupadas en 9 conjuntos, de forma tal que cada grupo de reglas se utiliza para la eliminación de un determinado tipo de sufijos, aplicándose los grupos de reglas en un determinado orden. En cada regla, se incluye, básicamente, una pareja de sufijos o terminaciones. Si se encuentra el primero de ellos en una palabra es sustituido por el segundo. La aplicación sucesiva de los grupos de reglas a una palabra, permite obtener el término en forma canónica que la representa. En la figura 3.5 aparece uno de los nueve grupos utilizados, con 21 reglas. Por ejemplo, la aplicación a la palabra "digitalization" de la regla 214, produce como resultado el término "digitalize". Otras reglas existentes en los siguientes grupos a aplicar, producen las transformaciones del término a "digital" (`303, "alize","al",..`), y finalmente a "digit" (`401,"al",LAMBDA,..`).

```
static RuleList step2_rules[] =
        {
          203,  "ational",   "ate",   6,  2,  0,  NULL,
          204,  "tional",    "tion",  5,  3,  0,  NULL,
          205,  "enci",      "ence",  3,  3,  0,  NULL,
          206,  "anci",      "ance",  3,  3,  0,  NULL,
          207,  "izer",      "ize",   3,  2,  0,  NULL,
          208,  "abli",      "able",  3,  3,  0,  NULL,
          209,  "alli",      "al",    3,  1,  0,  NULL,
          210,  "entli",     "ent",   4,  2,  0,  NULL,
          211,  "eli",       "e",     2,  0,  0,  NULL,
          213,  "ousli",     "ous",   4,  2,  0,  NULL,
          214,  "ization",   "ize",   6,  2,  0,  NULL,
          215,  "ation",     "ate",   4,  2,  0,  NULL,
          216,  "ator",      "ate",   3,  2,  0,  NULL,
          217,  "alism",     "al",    4,  1,  0,  NULL,
          218,  "iveness",   "ive",   6,  2,  0,  NULL,
          219,  "fulnes",    "ful",   5,  2,  0,  NULL,
          220,  "ousness",   "ous",   6,  2,  0,  NULL,
          221,  "aliti",     "al",    4,  1,  0,  NULL,
          222,  "iviti",     "ive",   4,  2,  0,  NULL,
          223,  "biliti",    "ble",   5,  2,  0,  NULL,
          000,  NULL,        NULL,    0,  0,  0,  NULL,
        };
```

Figura 3.5: Grupo de reglas de eliminación de sufijos.

3. Se obtiene un conjunto de términos de indexación con m elementos para la representación a partir de las palabras aparecidas en el manual, tras la aplicación de la lista de parada y el algoritmo de extracción de raíces. En concreto, $m = 6390$ términos para la sección 1 del manual de referencia. Para cada término $term_i$ del conjunto obtenido, se computa su frecuencia de documento (document frequency), df_i, y el peso del término en la colección, w_i, mediante la expresión:

$$w_i = log_2 (n / df_i)$$

en donde n es el número de documentos existentes en el manual. En concreto, $n = 524$ para la sección 1 del manual de referencia. En la figura 3.6 se incluyen ejemplos de los valores de las frecuencias de aparición en documentos y pesos, calculados para 7 de los 6390 términos de indexación utilizados. Puede también apreciarse como debido a un error de infrarradicación (understemming) del algoritmo de Porter, se han obtenido dos términos de indexación para la representación de un mismo significado: "copi" y "copy" (en concreto, a partir de las palabras "copy" y "copies" sí se obtiene "copi", pero a partir de "copying", se obtiene "copy").

4. Para cada documento d_j, se computa la frecuencia de aparición de cada término $term_i$ en él, tf_{ji}, y el peso del término en el documento, wd_{ji}, mediante la expresión:

$$wd_{ji} = tf_{ji} \cdot w_i$$

	df_t	w_t
chang	128	2.0362
copi	81	2.6963
copy	14	5.2288
file	316	0.7324
host	46	3.5126
permiss	48	3.4512
remot	34	3.9487

Figura 3.6: Frecuencias de aparición y pesos de términos en la colección.

	tf_{jt}			wd_{jt}		
	cp	*chmod*	*ftp*	*cp*	*chmod*	*ftp*
chang	0	12	4	0.0000	24.4341	8.1447
copi	23	0	0	62.0150	0.0000	0.0000
copy	1	0	0	5.2288	0.0000	0.0000
file	24	18	109	17.5774	13.1831	79.8308
host	0	0	11	0.0000	0.0000	38.6387
permiss	1	19	0	3.4512	65.5730	0.0000
remot	0	0	85	0.0000	0.0000	335.6404

Figura 3.7: Frecuencias de aparición de términos en documentos y pesos asociados a sus componentes en los vectores.

Cada documento queda representado, o indexado, por un vector de dimensión m, con los pesos asignados a cada uno de términos de indexación:

$$(wd_{j1}, wd_{j2}, ..., wd_{jm})$$

En la figura 3.7 aparecen los valores de las frecuencias de aparición, tf_{ji}, de los siete términos concretos anteriormente considerados, en los documentos correspondientes a 3 órdenes de las 525 totales: *cp* (copia de ficheros), *ftp* (transferencia de ficheros entre máquinas "remotas" en red) y *chmod* (cambio de los permisos de acceso, como los de lectura y escritura, a un fichero). Se incluyen también en la figura 3.7 los valores de las componentes de los vectores, wd_{ji}, correspondientes a estos términos de indexación.

5. Para el procesamiento de las consultas se utiliza la lista de parada y el algoritmo de extracción de raíces, obteniendo los términos de indexación aparecidos en ellas, de una forma análoga al de los documentos. Para una consulta q_k se calcula la frecuencia de aparición de cada termino de indexación $term_i$ en ella, tfq_{ki}, y se le asigna a cada término un peso en la consulta:

$$wq_{ki} = tfq_{ki} \cdot w_i$$

ftp.1c	0.0658169	csh.1	0.0318188
tip.1c	0.0595501	dos.1	0.0224357
make.1	0.0418136	List.1	0.0222939
rdist.1	0.0363193	chmod.1v	0.0206806
telnet.1c	0.0309496	make.1	0.0202500
rsh.1c	0.0292155	ftp.1c	0.0182902
rlogin.1c	0.0246991	mail.1	0.0181593
List.1	0.0246437	mailtool.1	0.0152526
rcp.1c	0.0223853	bar.1	0.0141064
cu.1c	0.0180467	ls.1v	0.0135503

(a) q_1 = copy a file from a remote host (b) q_2 = change the permission access to a file

Figura 3.8: Similitudes de documentos a consultas.

De esta forma, la consulta queda representada por un vector de dimensión m, con los pesos asignados a cada uno de términos de indexación:

$$(wq_{k1}, wq_{k2}, ..., wq_{km})$$

6. El valor de la similitud entre un documento d_j y una consulta q_k se obtiene mediante la fórmula del coseno del ángulo entre ambos vectores:

$$sim(d_j, q_k) = \frac{\sum_{i=1}^{m} wd_{ji} \cdot wq_{ki}}{\sqrt{\sum_{i=1}^{m} wd_{ji}^{2} \cdot \sum_{i=1}^{m} wq_{ki}^{2}}}$$

El cálculo de las similitud entre los documentos y una consulta determinada constituye la base para la selección de los documentos relevantes al usuario. Se calcula la similitud de todos los documentos con la consulta, se normalizan sus similitudes, de forma que su suma sea la unidad y se facilite su interpretación por el usuario, y se le presenta la lista de los nombres de los documentos ordenados decrecientemente por su valor de similitud.

3.4.2 Ejemplos de cálculo de similitud

En la figura 3.8 aparecen las similitudes de los documentos con dos consultas calculadas por el sistema. Estos datos son tomados de la salida directa del programa. Como ya se ha señalado, Superman resulta accesible tanto como subsistema (como se hace en Argos), como aplicación accesible desde el intérprete de órdenes. De esta forma, la inserción de las consultas de la figura 3.8 se puede hacer directamente por el usuario desde la línea de órdenes:

```
home/> superman copy a file from a remote host
home/> superman change the permission access to a file
```

Estas dos consultas han sido tomadas de un conjunto de ellas utilizadas para los experimentos descritos en el capítulo 5. Para las consultas se dispone de una asignación de documentos relevantes realizada por expertos humanos.

La consulta q_1 de la figura 3.8.a, "copy a file to a remote host", tiene como órdenes relevantes, asignadas por expertos, *rcp* y *ftp*. En la figura aparecen las similitudes de los diez documentos más relevantes calculadas por el sistema de la forma detallada en párrafos anteriores. La primera orden relevante calculada por el sistema, *ftp* es, por lo tanto acertada, mientras que las siete siguientes no. La orden *rcp* aparece en noveno lugar. La consulta q_2 de la figura 3.8.b, "change the permission access to a file" tiene como orden relevante, asignada por expertos, *chmod*. En este caso, la primera (y única) orden relevante se encuentra en la cuarta posición en el ordenamiento (ranking). Recordemos que la selección de estas órdenes la realiza el sistema sobre un conjunto de 525.

3.4.3 Estructura del sistema

El sistema Superman está compuesto a su vez por dos subsistemas. El primero de ellos, el subsistema *Indexación*, procesa todos los documentos de la base de datos, creando un fichero en el que se encuentran los términos de indexación y los documentos indexados. El segundo subsistema, *Similitud*, procesa las consultas y realiza el cálculo de la similitud de la consulta del usuario con los documentos existentes en la base de datos, generando la lista de los documentos ordenados por su relevancia. En la figura 3.9 incluimos los Diagramas de Flujo de Datos (DFDs) [Yourdon79, Rumbaugh91, Pressman93] de nivel 0 y 1 correspondientes a ambos subsistemas.

En el DFD0 del subsistema *Indexación* el elemento de datos *Manual* representa el conjunto de archivos de texto en el que se encuentra la documentación del manual. En el sistema operativo, existe un archivo de texto asociado a cada orden. En *ListaDocs* se encuentran los nombres de estos archivos, que serán los indexados. El elemento de datos *Indices* representa el archivo que se genera con el resultado del proceso de indexación, de acuerdo con las técnicas antes detalladas, y que está formado por el conjunto de términos de indexación, $term_i$, con sus pesos, w_i, y la representación de los documentos mediante sus vectores $(wd_{j1}, wd_{j2}, ..., wd_{jm})$.

En el DFD1 del subsistema *Indexación* se proporciona mayor detalle de cómo se realiza el proceso. A partir de los elementos de datos *ListaDocs* y *Manual* se obtiene el elemento de datos *DocTermFreq*, que incluye la lista de términos obtenidos tras la utilización de la lista de parada y la extracción de raíces, y su frecuencia de aparición en cada documento, tf_{ji}. A partir de estos datos, se obtienen los pesos de los términos w_i y los vectores que representan cada documento $(wd_{j1}, wd_{j2}, ..., wd_{jm})$, que se corresponden con el elemento de datos *DocVecTermW*. Por último, estos datos se graban en el archivo *Indices* en disco.

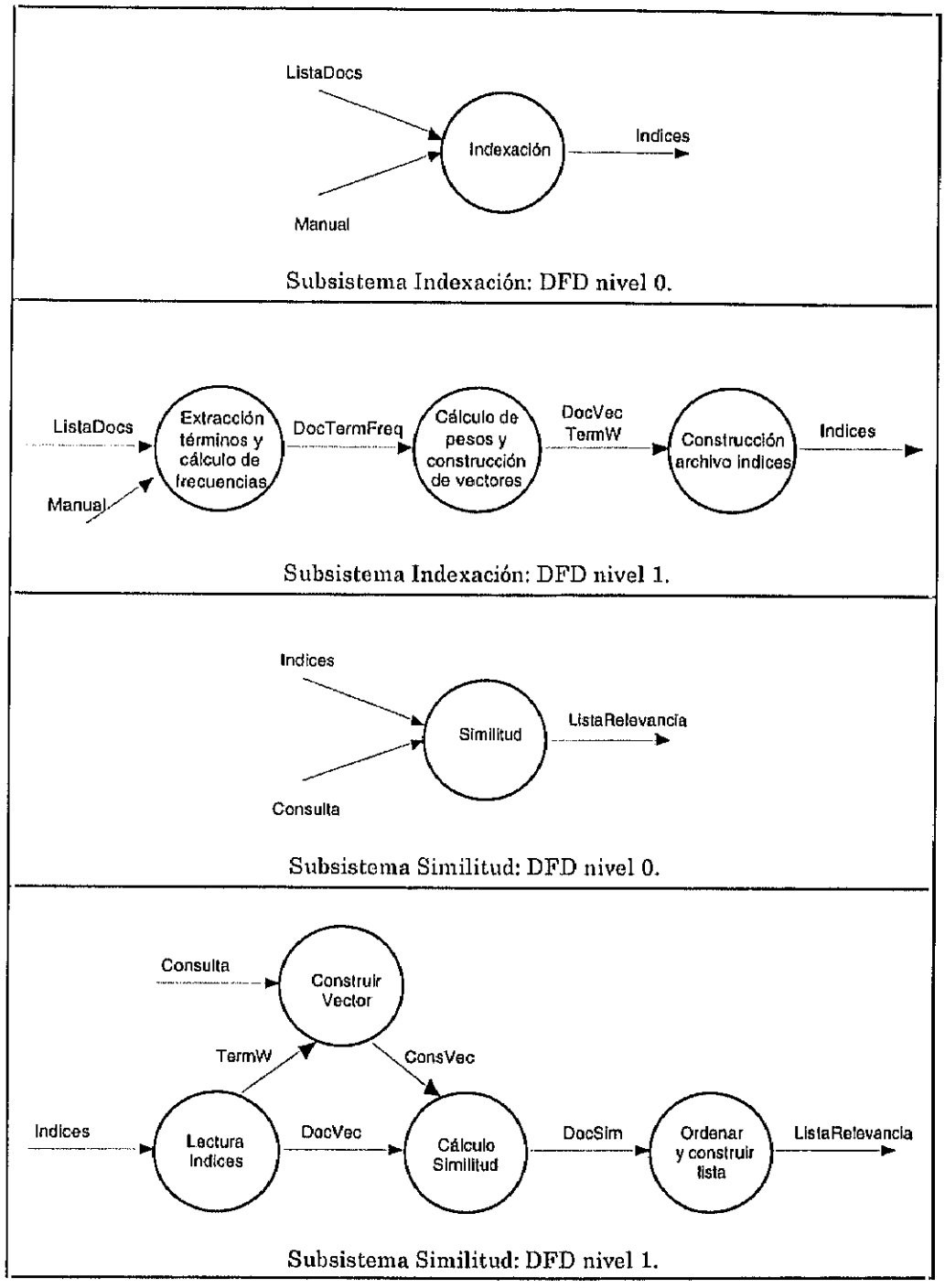


Figura 3.5: Diagramas de Flujos de Datos de nivel 0 y 1 de los subsistemas *Indexación* y *Similitud*.

En el DFD0 del subsistema *Similitud*, el elemento de datos *Indices* se corresponde con el archivo de indexación anteriormente citado, y el elemento de datos *Consulta* con una lista de caracteres almacenando la consulta que se utiliza para el cálculo de la similitud, que se le proporciona al sistema como argumento.

En el DFD1 del subsistema *Similitud*, el elemento de procesamiento *LecturaIndices* se ocupa de cargar los datos del archivo de índices proporcionando el elemento de datos *TermW*, correspondiente al conjunto de términos de indexación con sus pesos w_i y el elemento de datos *DocVec*, con los vectores de pesos de los documentos. A partir de *Consulta* y *TermW*, se obtienen los términos de indexación aparecidos en la consulta y los pesos asignados, generándose el vector $(wq_{t1}, wq_{t2}, ..., wq_{tm})$, que se incluye en el elemento de datos *ConsVec*.

El subsistema *Indexación* se ejecuta una vez mientras el conjunto de los documentos de la base de datos no se modifique. Su ejecución tiene un mayor coste computacional que la del sistema *Similitud*. En el entorno utilizado para su desarrollo, una estación de trabajo Sun Sparc-10, el procesamiento de los documentos de las 525 órdenes de la sección 1 del manual de referencia, con una longitud media de 762 palabras y un espacio de almacenamiento superior a los 2 Mbytes, conlleva un tiempo inferior a la decena de minutos. La ejecución de este subsistema no está orientada al usuario habitual, sino al administrador del sistema.

El subsistema *Similitud* se ejecuta una vez por cada procesamiento de una consulta. Su ejecución es menos costosa computacionalmente y conlleva un tiempo del orden de un segundo en la generación de la salida. La ejecución de este subsistema se encuentra al alcance de cualquier usuario y se puede realizar mediante el uso de la orden "superman" desde el intérprete de órdenes del S.O., proporcionando como argumentos la consulta en lenguaje natural. La salida del sistema se proporciona por la salida estándar del sistema operativo. De esta forma puede ser utilizado bien desde una aplicación (como el sistema Argos), bien directamente por el usuario. Esto último permite la utilización del sistema desde entornos no dotados de capacidades gráficas, como demanda Argos. El tiempo de ejecución del sistema es siempre inferior a los cinco segundos para consultas de longitudes menores de 1000 palabras.

Los dos subsistemas han sido implementados íntegramente en C en el entorno SunOS 4.1. sobre una estación de trabajo Sun Sparc-10. Detalles adicionales de implementación se pueden encontrar en la bibliografía [Díaz95].

3.5 Resumen y conclusiones

En este capítulo hemos presentado el sistema Argos, un sistema de ayuda para la recuperación de componentes en el sistema operativo UNIX, basado en la documentación y la formulación de consultas en lenguaje natural. Argos se ha definido con vistas a constituir un sistema práctico, efectivo, generalizable y que proporcione un marco de aplicación concreto orientado a la investigación.

De hecho, en la definición del sistema se han tenido presentes aspectos complementarios al procesamiento de textos y consultas, tales como criterios de interacción hombre-máquina e hipertexto. Por otra parte las ideas en que se basa el sistema han sido utilizadas para el desarrollo de sistemas similares sobre otras colecciones de componentes, como SmallTalk y VisualObject.

Las funcionalidades del módulo de Recuperación de Información del sistema Argos han sido implementadas por un subsistema basado en el modelo del espacio vectorial, utilización de pesos de términos, listas de parada y extracción de raíces.

Argos proporciona la definición del marco concreto de aplicación del sistema Ares, que se centra en el uso de técnicas de Procesamiento de Lenguaje Natural para la Recuperación de Información y que presentamos en un capítulo siguiente. Argos representa nuestro modelo de aplicación de las técnicas RI en bibliotecas de componentes software y proprociona la definición del dominio concreto sobre el que se centra nuestro trabajo: la documentación de componentes del S.O. UNIX, en concreto, de las órdenes disponibles desde su intérprete. Como veremos en capítulos siguientes, el uso de técnicas de Procesamiento de Lenguaje Natural en el sistema Ares permite mejorar la efectividad del proceso de recuperación respecto a una aproximación basada en el modelo del espacio vectorial, centrándose en determinados aspectos específicos de las descripciones de componentes software.

CAPITULO 4

EL PROCESAMIENTO DEL LENGUAJE NATURAL PARA LA RECUPERACION DE LA INFORMACION

4.1 Introducción

El Procesamiento del Lenguaje Natural (Natural Language Processing) tiene como objetivo la consecución de sistemas informáticos capaces de implementar funcionalidades que impliquen la comprensión, o la generación, de expresiones en lenguaje natural [Carbonell87]. Ejemplos de este tipo de sistemas lo constituyen los interfaces en lenguaje natural a bases de datos y sistemas expertos, sistemas de corrección inteligente de textos, sistemas de generación automática de resúmenes y sistemas de traducción automática. El Procesamiento del Lenguaje Natural (PLN) es un área en el que convergen trabajos desarrollados en disciplinas más amplias como la Inteligencia Artificial, la Lingüística Computacional y la Psicología Cognitiva [Carbonell87]. Ballard se refiere al Procesamiento del Lenguaje Natural como "*the engineering side of the discipline (computational linguistics)*" [Ballard87]. Sin entrar en la discusión que puede suscitar esta afirmación, nos sirve para centrar el enfoque de nuestro estudio. En este capítulo tratamos el PLN desde la vertiente de la ingeniería, o desarrollo de sistemas computacionales, centrándonos en aquellos sistemas más relevantes para el desarrollo de sistemas orientados a la Recuperación de la Información. Otros aspectos relacionados con modelos del lenguaje centrados en su vertiente lingüística o psicológica se referencian de forma superficial y siempre desde el enfoque del desarrollo de sistemas informáticos.

La aparición de los primeros sistemas orientados a la comprensión o generación del lenguaje natural, y enmarcados dentro del ámbito del PLN, no es reciente. Sistemas como BASEBALL, esencialmente un interfaz en lenguaje natural a una base de datos, aparecieron ya a principios de la década de los 60 [Ballard87]. Otros sistemas desarrollados posteriormente se han orientado a la implementación de capacidades de comprensión próximas a las humanas (lo que algunos autores han denominado *comprensión profunda* [Dyer83]), pero han mantenido siempre una constante condición de prototipo, en el sentido de que el subconjunto del lenguaje natural que procesan no es lo suficientemente amplio como para permitir la construcción de aplicaciones prácticas.

Ha sido en tiempos recientes cuando han aparecido una serie de sistemas que, sobre dominios concretos, han conseguido implementar una serie de funciones limitadas, pero cubiertas de forma suficiente como para permitir su aplicación práctica. Sistemas que se presentan como representantes de este hecho en nuestro estudio son los *sistemas de extracción de información*. Un aspecto común de este tipo de sistemas es el importante esfuerzo de desarrollo que conllevan [Hobbs92]. Una parte importante del avance experimentado en el campo se debe a la aparición de elementos orientados a facilitar el desarrollo de los sistemas. En nuestro estudio analizaremos la importancia de las grandes bases de conocimiento y de las *gramáticas de unificación*. Estas últimas constituyen la base de entornos altamente declarativos para el desarrollo de sistemas de PLN en los que una gran cantidad de detalles algorítmicos no necesitan ser especificados.

Croft distingue, básicamente, dos aproximaciones para integrar técnicas de Inteligencia Artificial en sistemas de RI: una fuerte y otra débil [Croft93]. La *aproximación fuerte* involucra un uso extensivo de capacidades de razonamiento. En esta aproximación se enmarcarían sistemas orientados a una compresión profunda que serían capaces de conseguir una interpretación completa, tanto de documentos como de consultas, y de proporcionar al usuario la información adecuada a sus necesidades. Esta aproximación, aunque ideal, resulta en la práctica infactible. Por otra parte, la mayoría de los modelos y sistemas desarrollados hasta el momento, los considera enmarcados dentro de la *aproximación débil*. En esta aproximación se utiliza un cierto grado de análisis sintáctico/semántico/contextual de documentos y/o consultas orientado a la mejora del proceso de recuperación, con menores demandas de capacidades de razonamiento.

Desde el punto de vista de la aproximación débil, la existencia de un esfuerzo investigador importante, orientado a la utilización de la información lingüística en los sistemas de RI que las técnicas y metodologías existentes en PLN han hecho posible, ha sido una constante histórica en ambos campos [Salton89b, Hess92, Croft93]. A este respecto, puede decirse que la utilización de métodos de extracción de raíces (stemming), que proporciona importantes mejoras en la efectividad, se puede considerar como la inclusión en los sistemas de RI de un restringido análisis lingüístico, a nivel morfológico. El avance del estado del arte en PLN ha conllevado la sucesiva aparición de modelos que incluyen un análisis lingüístico de niveles superiores, de textos y consultas en el proceso de recuperación.

En nuestro trabajo diferenciamos dos aproximaciones básicas [Salton89b, Hess92]: la *aproximación basada en la sintaxis* y la *aproximación basada en la semántica*. En estas dos aproximaciones pueden enmarcarse los sistemas de RI basados en el PLN que tratamos. Los sistemas que siguen la primera aproximación se basan en la utilización de la estructura sintáctica de las oraciones de los textos para mejorar la indexación de los mismos. Los sistemas que siguen la segunda aproximación se basan en criterios centrados en la utilización de información semántica y en el análisis del contenido de los textos.

La organización de este capítulo es como sigue. En primer lugar se tratan los niveles de descripción lingüística: morfológico, sintáctico, semántico y pragmático [Winograd83, Allen94]. Se analiza la información y los algoritmos que constituyen

la base del procesamiento correspondiente a cada uno de los niveles. En dos puntos siguientes se tratan las gramáticas de unificación y las grandes bases de conocimiento, desde el punto de vista de su utilización para el desarrollo de sistemas de PLN. En un punto siguiente se analizan los sistemas de PLN que hemos considerado especialmente relacionados con nuestra aproximación: interfaces en lenguaje natural, sistemas de comprensión de texto, sistemas de extracción de la información y sistemas de recuperación de información. Los dos últimos puntos siguientes tratan con mayor detalle los sistemas de RI utilizan técnicas de PLN, agrupándolos de acuerdo con la aproximación que siguen: sintáctica o semántica. Finalmente, se presenta un resumen y conclusiones.

4.2 Niveles de descripción lingüística

Tradicionalmente, se han identificado una serie de niveles de descripción del lenguaje que se pueden presentar en forma estratificada [Winograd83], comenzando por los niveles más próximos a la realización superficial (sonidos, frases escritas) y acabando por los más próximos a las capacidades cognitivas. En estos niveles se identifican elementos tales como fonemas, morfemas y palabras. La organización de cada nivel es relativamente independiente de los niveles inferiores. Por ejemplo, las reglas de organización de los sonidos para la formación de palabras son independientes en gran parte de las de organización de las palabras para la formación de frases. Esta estratificación es una organización natural para cualquier sistema complejo y puede ser utilizada como base para la de un sistema computacional [Winograd83]. Una posible clasificación de niveles sería la siguiente [Allen94, Cortés93]:

- *Nivel fonético y fonológico*: se tratan las relaciones existentes entre los sonidos y las palabras que representan. Este conocimiento es crucial para sistemas de reconocimiento y generación de voz pero no es relevante para nuestro estudio.
- *Nivel morfológico y léxico*: se trata del estudio de las palabras como unidades y la forma en que pueden estructurarse mediante unidades más básicas, los morfemas.
- *Nivel sintáctico*: se tratan las formas en que las palabras se pueden agrupar para formar oraciones. Se estudia el papel estructural que puede asignarse a las palabras y a unidades mayores que se denominan sintagmas.
- *Nivel semántico*: se estudian los significados de las palabras y cómo pueden combinarse para formar el significado de las oraciones. Este estudio se realiza de forma independiente del contexto, que es considerado en el siguiente nivel.
- *Nivel pragmático y de procesamiento del discurso*: se estudia la forma en que las oraciones se usan dentro de un contexto y cómo afecta ésto a la interpretación de la oración.

El uso de esta diferenciación de niveles como base para la organización de un sistema computacional tiene ventajas en cuanto a sencillez conceptual y modularidad del diseño. En la figura 4.1 se presenta una posible organización de un sistema de análisis basado en la estructuración en etapas secuenciales [Verdejo85]. Sin embargo, este modelo no es un modelo plausible de la forma de comprensión

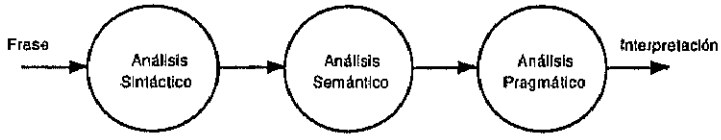


Figura 4.1: Sistema basado en etapas secuenciales

humana del lenguaje natural. Las personas utilizan diferentes recursos fonológicos, morfológicos, sintácticos, etc. sin un orden predeterminado, e incluso pueden llegar a entender el significado de un mensaje que no cumple ningún tipo de estructura conocida [Winograd83].

El uso de la anterior diferenciación de niveles en sistemas de PLN no ha sido uniforme. En los primeros sistemas, como BASEBALL, SIR o ELIZA [Ballard87], desarrollados en la década de los 60, esta estructuración de niveles apenas era considerada. En sistemas posteriores como TEAM [Bates87] es utilizada en mayor grado como base para la modularización del sistema. Una tendencia diferente, en la que se presta atención menor a la anterior diferenciación de niveles, constituye la representada por sistemas como BORIS [Dyer83] o DMAP [Martin89] que se basan en un modelo de la comprensión del lenguaje, la teoría de la dependencia conceptual [Schank72], y en un modelo de la memoria [Schank81, Schank82].

En cualquier caso esta diferenciación de niveles proporciona un marco adecuado para la discusión de los principales conceptos utilizados en la mayoría de los sistemas desarrollados [Allen94]. En los siguientes apartados tratamos las formas de procesamiento del lenguaje natural asociadas a cada uno de los niveles. Analizamos el conocimiento o la información utilizados en cada nivel y las propuestas para su formalización, así como los problemas planteados en cada uno de ellos. Destacamos los conceptos más relevantes para nuestro trabajo, especialmente aquellos que son utilizados en los sistemas descritos en puntos posteriores de este capítulo, y en el siguiente. Los niveles fonológico y fonético no se tratarán por no ser relevantes a nuestro estudio. La estructuración general de este punto sigue la línea adoptada en por la mayoría de los textos incluidos en la bibliografía [Allen94, Rich91, Beardon91, Cortés93, Rodríguez94].

En cada nivel diferenciamos dos cuestiones: la información utilizada para el análisis y las técnicas de procesamiento, haciendo referencia con estas últimas a los detalles de implementación de los algoritmos utilizados en este nivel. Para modelos de procesamiento concretos, como por ejemplo, las gramáticas semánticas, resulta difícil separar ambas cuestiones e incluso distinguir los niveles de descripción. A pesar de ello, la aproximación seguida aquí resulta especialmente interesante con vistas a la implementación de sistemas mediante gramáticas de unificación, tratadas en un punto siguiente de este capítulo.

4.2.1 Nivel léxico y morfológico

La información léxica

Mediante el término *léxico* (*lexicon*) se hace referencia a la componente del sistema que almacena información asociada a las palabras. En la mayoría de los sistemas actuales, esta información incluye propiedades de las palabras de las que se hace uso en otros niveles. Allen propone en concreto la inclusión de la siguiente información en el léxico [Allen94]:

- *Categorización sintáctica*. Se trata de etiquetado dependiente del tipo de formalismo sintáctico utilizado. Existen categorías cerradas (e.g. preposición, determinante) y otras abiertas (e.g. nombre, adjetivo).
- *Propiedades sintácticas de concordancia*, como el género, el número, la persona o el caso.
- *Otras propiedades sintácticas*, como las restricciones selectivas (e.g. tipo de argumentos que un verbo admite) o el tipo de preposiciones que una palabra admite.
- *Información morfológica*, como el patrón de formación de la palabra.
- *Información semántica*, dependiente del modelo de tratamiento semántico utilizado.

En párrafos siguientes trataremos con más detalle la información léxica utilizada en cada nivel y que puede ser incluida en el léxico. Las entradas del léxico, o ítems léxicos, no tienen por qué coincidir con las palabras ortográficas: pueden existir entradas determinadas por las raíces de las palabra, por cada uno de sus significados (polisemia/sinomia), o por patrones formados por varias palabras [Verdejo94].

El procesamiento léxico

El procesamiento léxico consiste esencialmente en la identificación de los ítems léxicos aparecidos en el texto de entrada. La no coincidencia de los ítems léxicos con las palabras ortográficas conlleva una fuerte relación del procesamiento léxico con el morfológico (que trataremos a continuación).

En el caso de sistemas comerciales para el tratamiento de textos en formato electrónico [Hobbs92] se realizan una serie de labores de preprocesamiento de la documentación, previas al procesamiento léxico, que se encuentran también fuertemente relacionadas con este último. En estas labores de preprocesamiento, las cuestiones relacionadas con la eficiencia juegan un papel importante, y resulta adecuada la utilización de algoritmos de análisis basados en autómatas finitos [Johnson68, Aho85]. La herramienta *lex* del sistema operativo UNIX, constituye un ejemplo paradigmático para la implementación eficiente de este tipo de analizadores [Sun88, Coffin95].

La información morfológica

La morfología se ocupa del estudio de la estructura y formación de las palabras. Los morfemas constituyen la unidad fundamental en este nivel. En lenguas como el Inglés, la clase y complejidad de los problemas que se plantean es menor que en otras como el Castellano o el Francés, y éstas a su vez presentan una menor complejidad que algunas otras como el Finlandés o el Vasco [Verdejo94].

La información lingüística utilizada en este nivel incluye los morfemas del lenguaje analizado y las reglas de formación de las palabras. En Inglés se diferencian morfemas raíces y morfemas afijos. Los afijos se categorizan en prefijos y sufijos. Las reglas de formación pueden codificarse así [Beardon91]:

```
IF
    la palabra finaliza en "y" AND la penúltima letra no es vocal
THEN
    el plural se forma:
        suprimiendo la letra "y" y
        añadiendo "ies"
```

Una lista detallada de estas reglas para el Inglés se puede encontrar en [WN93]. Las reglas de formación se apoyan en la categorización de las palabras en tipos conocidos como *partes del discurso* (*parts of speech*). En Inglés estas son: nombres, pronombres, determinantes, adjetivos, preposiciones, verbos, adverbios y conjunciones [Beardon91]. Esta información puede incluirse en el léxico para posibilitar el análisis morfológico.

Las reglas de formación morfológicas también pueden expresarse mediante gramáticas formales. El tipo de gramática que es necesario utilizar depende del idioma. Como ejemplo, para el Inglés la información morfológica puede ser representada adecuadamente mediante una gramática regular, mientras que para el Italiano puede resultar necesaria una independiente del contexto [Antonacci89].

El procesamiento morfológico

Dependiendo de las reglas de formación, pueden resultar adecuados algoritmos basados en autómatas de estados finitos o autómatas a pila. Estos últimos pueden ser a su vez algoritmos descendentes o ascendentes. Tratamos estas cuestiones en el siguiente punto, al hablar de los algoritmos de análisis en el nivel sintáctico, por ser de mayor relevancia en este nivel y ser lo habitual en la bibliografía [Allen94, Beardon92, Cortés93, Rich91].

4.2.2 El nivel sintáctico

En el nivel sintáctico se estudian las formas de agrupación de las palabras en la construcción de oraciones, atendiendo generalmente a su orden de aparición.

La información sintáctica

La información utilizada en este nivel puede representarse, en una primera aproximación, por una gramática independiente del contexto en la se utilizan como símbolos no terminales las partes del discurso, y otras tales como frase nominal o frase verbal; como símbolos terminales las palabras que constituyen el léxico del lenguaje. Una gramática sencilla de este tipo sería [Allen94]:

```
S ->    NP VP                ART ->    the
NP ->   ART N                ADJ ->    big
NP ->   ART ADJ N            V ->      eats
VP ->   V                    N ->      man
VP ->   V NP                 N ->      apple
```

Se pueden determinar restricciones adicionales existentes en los lenguajes naturales que también se enmarcan en el nivel sintáctico, tales como la concordancia de número o de género entre categorías gramaticales. Esta información puede representarse adecuadamente mediante *gramáticas de atributos* (*attribute grammars*), introducidas inicialmente por Knuth [Knuth68, Allen94]. Las gramáticas de atributos permiten la inclusión de atributos en los símbolos gramaticales, y de ecuaciones o funciones basados en ellos en las reglas. Un ejemplo de regla en la que se impone la concordancia de número mediante el uso de un atributo denominado así en este tipo de gramáticas sería:

```
NP ->    ART N ( ART.num = N.num )
```

Restricciones del tipo de la anterior pueden ser representadas también mediante una gramática libre de contexto que haga uso de reglas y símbolos adicionales. La capacidad de representación de las gramáticas independientes del contexto para la sintaxis del LN es una cuestión aún en debate entre la comunidad lingüística [Gazdar89], en cualquier caso, el uso de los atributos para la representación de restricciones lingüísticas como las anteriormente mencionadas facilita la redacción de la gramática y mejora su legibilidad. Esta cuestión se trata en un punto siguiente de este capítulo al tratar las gramáticas de unificación.

El procesamiento sintáctico

La base de los algoritmos utilizados para el análisis sintáctico se fundamenta en los algoritmos orientados a gramáticas independientes del contexto y gramáticas de atributos desarrollados a finales de los 60 y durante la década de los 70 en torno a la implementación de compiladores [Allen94]. Una descripción detallada de este tipo de técnicas de análisis de compiladores se puede encontrar en [Aho85].

Cabe distinguir dos tipos de algoritmos de análisis, el descendente y el ascendente. En el análisis ascendente, se parte de los símbolos de cadena de entrada y se identifican las categorías gramaticales que pueden ser utilizadas para conseguir construir el símbolo inicial. En el análisis descendente se parte del símbolo inicial y se aplican las reglas de formación hasta conseguir obtener la cadena de entrada. Los analizadores ascendentes proporcionan una mayor eficiencia en la mayoría de los casos. El análisis basado en charts (*chart-parsing*), basado en el algoritmo de

Early [Aho85] utiliza una estrategia mixta mediante la que se consigue una mayor eficiencia en el análisis [Aho85, Allen94].

Los autómatas asociados a las gramáticas pueden representarse mediante diagramas de transición de estado. Las *Redes de Transición Recursivas* (*Recursive Transition Network, RTN*) constituyen una forma gráfica de representación de los lenguajes independientes del contexto. En una RTN los arcos que conectan los estados pueden estar etiquetados por símbolos terminales o por símbolos no terminales. Para atravesar un arco etiquetado por un no terminal se debe reconocer, o atravesar, el subgrafo correspondiente a ese no terminal. Una RTN es equivalente al autómata asociado a la gramática [Gazdar89]. Una *Red de Transición Aumentada* (*Augmented Transition Network, ATN*) es una RTN con la capacidad de comprobar condiciones y realizar acciones al atravesar sus arcos [Woods87, Gazdar89]. Allen presenta el trabajo desarrollado en torno a las gramáticas de atributos de Knuth, de uso frecuente en compiladores, como la base de las ATN [Allen94]. Los analizadores basados en ATN han sido quizá los más utilizados para la implementación de sistemas de PLN durante la década de los 70 y parte de los 80 [Gazdar89], tales como LIFER o LUNAR [Ballard87].

4.2.3 El nivel semántico

En la bibliografía [Schank77, Schank81, Schank82, Gazdar89, Riesbeck89, Thayse91, Allen94] se pueden encontrar diferentes modelos para el tratamiento semántico. Allen en concreto propone incluir en este nivel la interpretación semántica de las oraciones, sin incluir información contextual y diferencia en este nivel dos cuestiones fundamentales: la representación del significado y el proceso de interpretación. En este punto diferenciaremos, al igual que en anteriores, la información utilizada en este nivel y las técnicas de procesamiento o implementación. Además, de acuerdo con Allen, destacamos el problema de la caracterización del *lenguaje de representación del significado* (*Meaning Representation Language, MRL*), utilizando el término de uso frecuente en la bibliografía [Winston84, Gazdar89].

La representación del significado

Como formalismos semánticos o formalismos para la representación del significado, se propone toda la variedad de representaciones que se utilizan en los sistemas basados en el conocimiento [Barr81, Frost89, Verdejo94]. En una primera aproximación se suelen clasificar los formalismos de representación del significado en dos grandes familias: los basados en la lógica y los basados en estructuras que soportan modos de inferencia específicos, como las redes semánticas o los lenguajes de frames [Barr81, Frost89, Fernández-Valmayor91, Binot91, Thayse91, Cortés93, Verdejo94]. En general, existe un cierto acuerdo en concluir que las diferentes notaciones proporcionan facilidades para la representación de determinados tipos de información, pero todas admiten ampliaciones de forma tal que su poder expresivo, o de representación, sea el mismo [Binot91, Cortés93]. Es decir, resulta posible la representación de las construcciones lógicas en diferentes notaciones tales como las orientadas a redes o grafos, y viceversa [Binot91].

Trataremos a continuación tres formas de representación que puede presentarse como las más utilizadas para la construcción de sistemas de PLN [Allen94] y que son especialmente relevantes a nuestro estudio: introducen conceptos que serán utilizados en este y en siguientes capítulos.

Marcos de casos (case frames)

La representación mediante *marcos de casos* (*case frames*) se basa en la teoría de gramáticas de casos de Fillmore [Bruce87].

La noción lingüística tradicional de caso hace referencia a la clasificación de los nombres con respecto al papel sintáctico que desempeñan en una oración, determinado por su declinación o sufijo. Estos casos se suelen denominar "superficiales" o "sintácticos". Lenguajes naturales como el latín, ruso o finés presentan una estructura de casos de este tipo [Beardon91].

Alternativamente, se propone otro concepto de caso, que se define como la categorización de los sintagmas nominales de acuerdo con los papeles conceptuales que juegan en la acción desarrollada en una frase. Se trata de representar el significado de una frase por medio de la acción que en ella se describe, en la que los sintagmas nominales representan características de dicha acción. Estos roles semánticos se han dado en llamar *casos profundos* o *semánticos* (*deep cases*) [Bruce87].

En la bibliografía [Beardon91, Winograd83, Bruce87] se pueden encontrar diversas propuestas de sistemas de casos que difieren, esencialmente, en el número y nombre los de casos, que dan cuenta de diferentes *papeles* o *roles temáticos* identificados (*thematic roles*), como el de Simmons, el de Chafe [Winograd83] y el de Grimes [Bruce87]. El sistema de casos propuesto inicialmente por Fillmore incluía los siguientes casos o roles temáticos [Bruce87]:

- Agente (agent): Instigador de la acción.
- Contra-agente(counter-agent): Fuerza o resistencia contra la que se realiza la acción
- Objeto (object): Entidad acerca de cuya existencia, movimiento o cambio se refiere la acción
- Resultado (result): Entidad que comienza a existir como resultado de la acción
- Instrumento (instrument): Estímulo o causa física inmediata de un suceso
- Fuente (source): Lugar desde el que algo se mueve
- Objetivo (goal): Lugar hacia el que algo se mueve
- Paciente (experience): Entidad que recibe, acepta, experimenta o sufre los efectos de una acción

Para la representación en forma de frame basada en casos, los roles del frame, o atributos del marco, vienen dados por los nombres de los casos y los *fillers*, o valores de los atributos, por los nombres y modificadores de las frases nominales correspondientes. La cabeza (*head*) del marco viene dada por la acción, que juega un papel central en esta representación. El siguiente es un ejemplo de representación basada en un conjunto de casos parecido al anterior [Carbonell87]:

"In Elm Street, John broke a window with a hammer for Billy."

```
[BREAK
    [case frame
        agent: JOHN
        object: WINDOW
        instrument: HAMMER
        recipient:
        directive:
        locative: ELM STREET
        benefactive: BILLY
        co-agent: ]
    [modals
        time: past
        voice: active ]]
```

En ella se incluye también información adicional relativa al tiempo y a la voz de la expresión.

Estructuras de Dependencia Conceptual

La teoría de la dependencia conceptual fue concebida inicialmente por Schank como una teoría de la comprensión humana del lenguaje, orientada al procesamiento del Lenguaje Natural [Schank72]. La teoría de la dependencia conceptual postula que es posible representar el significado de un texto en forma canónica e independiente del lenguaje, de forma tal que frases diferentes, incluso en idiomas distintos, con igual significado tienen la misma representación. La teoría proporciona un MRL basado en las denominadas estructuras de dependencia conceptual. Una estructura conceptual es definida como una *red* o *diagrama de conceptos* (*Conceptual Diagram, CD*), en donde diferentes clases de conceptos (categorías conceptuales), se pueden encontrar relacionados de formas específicas (dependencias conceptuales) a otras clases de conceptos, en una forma próxima a las redes semánticas [Hardt87].

Inicialmente se definen cuatro clases de conceptos o categorías conceptuales [Schank72]:

- Acciones, la mayoría de los verbos (ACT, actions)
- Nominales (PP, picture producers,)
- Modificadores de nominales (PA, picture aiders)
- Modificadores de acciones (AA, action aiders).

Las acciones juegan un papel predominante en la teoría. En la teoría se introducen una serie de acciones primitivas. Se postula la posibilidad de representar el significado de cualquier acción a partir de la composición de estas acciones primitivas. Algunas de ellas son:

- ATRANS: significa transferencia de posesión, control o propiedad: dar, ceder, etc.
- PTRANS: significa cambio de posición: andar, volar, caer, etc.
- PROPEL: la aplicación de una fuerza: empujar, lanzar, etc.
- MTRANS: el intercambio de información: ver, oír, recordar.

El número de estas acciones primitivas ha ido evolucionando con el tiempo de forma paralela al desarrollo de sistemas basados en la teoría [Birnbaum81, Hardt87, Fernández-Valmayor91]. Como caso concreto, en la implementación del sistema BORIS [Dyer83] se utilizan 11 acciones primitivas:

ATRANS	PROPEL	PTRANS	INGEST	GRASP	ATTEND
EXPEL	MTRANS	MBUILD	MOVE	SPEAK	

En la teoría se incluyen una serie de reglas que especifican las diferentes formas en las que las categorías conceptuales pueden combinarse. Se proporciona una notación gráfica para la representación en los CD que en trabajos posteriores es sustituida por una equivalente basada en marcos (frames) [Hardt87, Fernández-Valmayor91]. Un ejemplo de representación en forma de CD sería el siguiente [Carbonell87]:

"John gave Mary a ball"

```
[ATRANS
     rel: POSSESSION
     actor: JOHN
     object: BALL
     source: JOHN
     recipient: MARY]
```

Representaciones basadas en la lógica

La lógica de primer orden ha constituido la base para la definición de un lenguaje de representación del significado en muchos de los sistemas referenciados en la bibliografía, como LUNAR o SHRDLU [Gazdar89]. También se ha propuesto el uso de lógicas de orden superior y de lógicas modales para utilización como MRL [Parikh87, Gazdar89, Delsarte90, Allen94].

Desde el punto de vista del desarrollo de sistemas de PLN, las notaciones más utilizadas han sido las basadas en la lógica de primer orden [Binot91]. En estas aproximaciones se utilizan las construcciones lógicas básicas, como los cuantificadores existencial y universal, y se introducen ampliaciones, como las que facilitan la representación del significado de los existentes en el lenguaje natural (e.g. "all", "some", "most", "many", "a few", "the", etc., en Inglés). Un ejemplo de notación de este tipo es la introducida por Allen [Allen94], que toma sus ideas iniciales de la utilizada para la implementación del sistema LUNAR [Carbonell87].

En la notación de Allen se incluyen también elementos basados en la idea de casos o roles temáticos, así como operadores para la representación del modo de las oraciones (ASSERT, Y/N-QUERY, COMMAND, WH-QUERY). El siguiente es un ejemplo de representación en esta notación.

What did the man eat?

```
(WH-QUERY (<PAST EAT> e1
     [AGENT <THE m1 MAN>]
     [THEME <WH w1 PHYSOBJ>]))
```

La información semántica

Para conseguir obtener la representación semántica de las oraciones los sistemas utilizan diferentes tipos de información. Un número importante de sistemas se basan en información contextual, que tratamos en el siguiente nivel. En este punto tratamos la información que no hace referencia al contexto de las oraciones.

La información utilizada en la implementación de un gran número de sistemas de PLN se ha basado en las ideas de restricciones seleccionales, propuestas originalmente por los lingüistas Katz y Fodor en los años 60 [Gazdar89]. Las restricciones seleccionales fueron introducidas como forma de dar cuenta del hecho de que una oración en conjunto puede ser no ambigua aunque las palabras individualmente pueden tener diferentes significados. La teoría se encuentra basada en la idea de que es posible asociar marcadores semánticos a cada significado de una palabra, especificando atributos del significado así como condiciones de los atributos de los significados que pueden combinarse con él.

Como ejemplo de información de este tipo, un significado posible de la palabra "spirit" (licor) podría tener el marcador semántico "objeto físico". Mientras que otro significado (espíritu), no. El adjetivo "yellow", para uno de sus significados (color), puede requerir que el nombre que califica tenga el marcador "objeto físico". Otro significado de "yellow" (cobarde) puede requerir que el nombre tenga el marcador "animado". De esta forma la frase "yellow spirit" sólo puede tener un significado (licor amarillo) de los cuatro posibles en este ejemplo.

Este tipo de información se ha adaptado al lenguaje de representación utilizado en cada sistema, prestándose a su integración en jerarquías conceptuales almacenadas en redes semánticas [Allen94]. Esta información semántica, asociada a los significados de las palabras, se almacena en gran parte en el lexicón de los sistemas. Como caso concreto, el sistema DYPAR [Dyer83, Martin89], uno de los analizadores semánticos más evolucionados de entre los basados en la teoría de la dependencia conceptual, dispone de la siguiente información en el léxico para uno de los significados de la palabra "give":

```
(WORD GIVE DEF (ATRANS
        ACTOR X <== (EXPECT-HEAD 'HUMAN'BEF)
        OBJECT * <== (EXPECT-HEAD 'PHYSOBJ 'AFT)
        TO * <== (EXPECT-HEAD 'HUMAN 'AFT)
        FROM X )
```

Mediante ella, se representa la necesidad de que para un significado de la palabra "give", el CD que represente el significado de la oración tendrá como acción primitiva ATRANS, el valor (filler) del role ACTOR deberá ser de tipo HUMAN, el de OBJECT de tipo PHYSOBJ, el de TO de tipo HUMAN, y el de FROM coincide con el de ACTOR. Adaptaciones similares se realizan en los sistemas basados en marcos de casos y representaciones lógicas [Allen94].

Información basada en el orden de aparición de los términos en las oraciones se utiliza para la interpretación semántica de la oración. Este tipo de información es el principalmente utilizado por los sistemas basados en el uso de patrones (*pattern*

matchig), como ELIZA [Carbonell87], en gramáticas semánticas, como LUNAR o LIFER [Burton87] y en los basados en un conjunto de reglas de transformación de la estructura sintáctica a la semántica [Allen94]. Parte de esta información puede incluirse también en el léxico, como en el ejemplo de la entrada vista anteriormente de DYPAR, en el que se incluye una serie de restricciones relativas a la posición de los términos en la oración: los términos que describen al actor deben encontrarse delante del verbo, así como los correspondientes al objeto deben encontrarse detrás.

El procesamiento semántico

Los algoritmos utilizados para la interpretación semántica son aún más diversos que los existentes para el procesamiento sintáctico. Se han desarrollado sistemas que siguen estrategias generales de tipo ascendente o descendente, y además una serie de aproximaciones altamente específicas o "ad-hoc", tal como la seguida en DYPAR, concebido como un analizador conducido por expectativas basado en "demonios" (*daemons*) [Dyer83], o el analizador basado en casos propuesto en DMAP [Martin89]. Para nuestro estudio, los detalles de implementación de estos algoritmos, juegan un papel secundario frente a los aspectos tratados en párrafos anteriores, ya que los formalismos gramaticales de unificación permiten la implementación de analizadores de forma declarativa, sin necesidad de detallar los algoritmos de análisis [Shieber89, Gazdar89].

4.2.4 El nivel pragmático

En el nivel pragmático se utiliza información contextual para la resolución de ambigüedades y la interpretación de una secuencia de oraciones. Las ideas sobre *guiones* (*scripts*), *planes* (*plans*) y *objetivos* (*goals*) propuestas por Schank y Abelson [Schank77] y las de los diálogos orientados a tareas y estructura del discurso propuesta por Grosz [Scha87] han sido las más utilizadas en la implementación de sistemas de PLN [Allen94]. Entre estos, el concepto de guión o script es el que tiene mayor relevancia para nuestro estudio.

Un script es un tipo específico de estructura de conocimiento utilizado para representar secuencias de eventos estereotipados. Existe una secuencia prototípica de eventos correspondiente al hecho de ir a comer a un restaurante, comprar un objeto en una tienda, ir a ver al doctor, un accidente de tráfico etc. El script se utiliza para conducir el proceso de comprensión, emitiendo predicciones acerca de posibles acciones y completando el significado por la aplicación de las inferencias que en él aparecen indicadas [Schank81].

Un script correspondiente a ir a un restaurante a comer puede almacenar una información como la siguiente [Gazdar89]:

```
script(restaurant(Customer, Restaurant, Food),
    [
        enters(Customer, Restaurant),
        calls(Customer, Server),
        brings(Server, Food),
        eats(Customer, Food),
        pays(Customer, Teller),
        leaves(Customer, Restaurant)
    ]) :-
        human(Customer),
        human(Server),
        human(Teller),
        restaurant(Restaurant),
        food(Food).
```

Utilizando esta información se pueden resolver las ambigüedades existentes en textos que traten acerca de este tipo de eventos. Por ejemplo, en un texto sencillo en que se describe un suceso del tipo "visita a restaurante" como el siguiente y cuyo comienzo sea:

"John went to a Pizzahut. He called Peter. He brought a lasagna."

En este caso se puede resolver la ambigüedad introducida por el uso del pronombre "he" en la segunda y la tercera oración (*anáfora*) utilizando la información del script anterior: el pronombre "he" en la segunda hace referencia a John ya que él es el agente de la primera oración (Customer), mientras que en la tercera oración "he" hace referencia a Peter (Server). Para este ejemplo, el script instanciado sería:

```
[   enters(john, pizzahut),
    calls(john, peter),
    brings(peter, lasagne),
    eats(john, lasagne),
    pays(john, peter),
    leaves(john, pizzahut)   ]
```

De esta forma, los scripts pueden ser utilizados también como forma de representación del significado (como los marcos o las estructuras de dependencia conceptual), pero a nivel de texto, no de oración, utilizándose un script para representar el significado de un texto completo. Por ejemplo, en un sistema basado en scripts, FRUMP, se utilizan scripts para la interpretación de noticias en formato electrónico. FRUMP dispone de unos 50 scripts diferentes, correspondientes a terremotos, secuestros, visitas diplomáticas y huelgas entre otros. En FRUMP interpretar una noticia es encontrar el script correspondiente e instanciar de forma adecuada los valores de sus atributos [Lehnert87]

Al igual que en el nivel semántico, las formas de implementación concretas y los algoritmos utilizados en los analizadores varían de unos sistemas a otros [Cullingford81, Dyer83, Schank89]. Al igual que para los niveles anteriores, las gramáticas de unificación permiten representar la información correspondiente a estructuras como los scripts de forma declarativa, sin necesidad de detallar los algoritmos de análisis [Shieber89, Gazdar89].

4.3 Gramáticas de unificación

Durante la década de los ochenta han hecho aparición un conjunto de formalismos gramaticales denominados en conjunto *formalismos gramaticales de unificación* (*unification grammars*) [Allen89, Shieber89, Shieber92].

Dos de los primeros formalismos de unificación introducidos fueron las *Gramáticas de Unificación Funcionales* de Kay (*Functional Unfication Grammar, FUG*), desarrollada a partir de las Redes de Transición Aumentadas (ATN), y las *Gramáticas Léxico Funcionales* de Bresnan y Kaplan (*Lexical-Functional Grammar, LFG*), con una orientación esencialmente lingüística [Shieber89, Smolka92].

En general, el conjunto de formalismos de este tipo desarrollados [Allen89, Shieber89, Shieber92, Gazdar89, Smolka92, Rodríguez94] puede presentar un aspecto bastante heterogéneo, pero de acuerdo con Smolka [Smolka92], opinamos que dichos formalismos consisten básicamente en un conjunto de reglas de estructura de frase más un conjunto de restricciones lógicas, gracias a las cuales consigue aumentarse la capacidad expresiva de las reglas incontextuales que les sirven de base, permitiendo además la especificación del análisis resultante de las expresiones de entrada.

El principal interés de las gramáticas de unificación para nuestro estudio reside en que constituyen la base de entornos de muy alto nivel, extremadamante declarativos, para el desarrollo de sistemas de PLN, y que facilitan en gran medida su implementación. En palabras de Shieber [Shieber92]:

> *"By utilizing declarative constructs that emphasize the modularization of information and its manipulation in the face of partiality, many technical problems in language description and computer manipulation can be solved."*

De esta forma, la implementación de un sistema de PLN se puede llevar a cabo mediante una especificación en gran medida declarativa, sin necesidad de detallar aspectos de implementación relativos al algoritmo de análisis utilizado (botton-up, top-dow, chart parser, etc.) Además contribuyen a una mayor modularización de la información que se utiliza para el análisis. Desde un punto de vista más general, podemos establecer un paralelismo [Gazdar89] con la idea de utilizar una representación explícita de las reglas en sistemas de IA, que ha demostrado su éxito en tareas como diagnostico médico o prospección geológica [Barr89, Rich91].

PATR y DCG son los dos formalismos más relevantes para nuestro enfoque. Sobre ambos formalismos se encuentran disponibles herramientas que permiten la implementación de sistemas [Shieber89]. PATR y PATR-II pueden presentarse como un estándar entre los formalismos gramaticales basados en rasgos [Gazdar89, Hirsh88, Rodríguez94]. Por otra parte, las *Gramáticas de Cláusulas Definidas* (*Definite Clause Grammar, DCG*), introducidas por Warren y Pereira [Pereira 80], constituyen el formalismo de este tipo más adecuado para el desarrollo rápido de sistemas simples y eficientes de PLN [Shieber 89].

En PATR y PATR-II se utilizan dos operaciones básicas: concatenación y unificación. La concatenación es la única operación permitida que combina cadenas. La unificación es la única operación permitida que combina información. La unificación puede utilizarse para la comprobación de restricciones contextuales y para elaboración del análisis, lo que conlleva que el formalismo sea completamente declarativo [Shieber89]. En PATR se añaden restricciones en forma de ecuaciones a las reglas gramaticales. Estas reglas gramaticales pueden representarse en forma de estructuras de rasgos, o, de forma equivalente, mediante *Grafos Acíclicos Dirigidos* (*Directed Acyclic Graphs, DAGs*), cuyos arcos llevan como identificadores los nombres de los rasgos [Shieber89, Gazdar89]. El resultado del análisis es el de la unificación de las estructuras de rasgos, o de los DAGs, correspondientes a los símbolos de la cadena a analizar y a las reglas gramaticales [Shieber89]. El siguiente sería un ejemplo sencillo de reglas gramaticales representadas en este formalismo:

```
Rule S -> NP VP
     <NP arg0> = <VP arg0>
     <arg0> = <NP arg0>
     <sem> = <VP sem>.
Word runs:
     <cat> = VP
     <arg0 per> = 3
     <arg0 num> = sing
     <sem> = run.
```

```
Word he:
     <cat> = NP
     <arg0 per> = 3
     <arg0 num> = sing
     <arg0 gen> = masc.
Word she:
     <cat> = NP
     <arg0 per> = 3
     <arg0 num> = sing
     <arg0 gen> = fem.
```

En ellas se define una regla de construcción de oraciones (*S*): un sintagma nominal seguido de uno verbal. Ambos sintagmas deben concordar en persona, número y género (atributo *arg0*). Se incluye también en la regla el atributo *sem*, orientado a representar el significado de la oración, que está determinado por el del sintagma verbal. Se define un léxico formado por tres palabras para las que se incluyen su categoría sintáctica y valor para sus atributos de persona y número. Para los nombres se incluye también su género, mientras que para el verbo se incluye su semántica. El valor del atributo *arg0* de *S* unifica con los de *NP* y *VP*, el del atributo *sem* de *S* con el de *VP*. En la figura 4.2. se presentan los DAGs correspondientes a dos palabras; así como el de su unificación mediante la regla de formación *S*, correspondiente al análisis de una oración con estas dos palabras.

Las Gramáticas de Cláusulas Definidas (DCG) fueron introducidas por Warren y Pereira [Warren80] como un potente formalismo para el cual Prolog proporciona un mecanismo eficiente de análisis, implementación de restricciones semánticas y traducción. En dicho trabajo, las DCGs se introducen como una alternativa al que había sido hasta entonces el formalismo más empleado en procesamiento del lenguaje natural, las ATN, argumentándose que las DCG son un formalismo tan eficiente como las ATN y más declarativo y potente. Desde el punto de vista lógico, procesar una oración mediante una DCG es equivalente a demostrar que una secuencia de palabras constituye una sentencia bien formada con respecto a una sintaxis y a unas restricciones contextuales concretas.

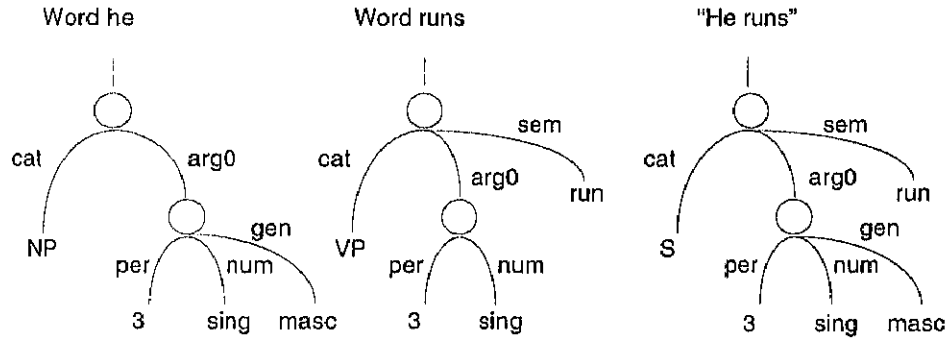


Figura 4.2: DAGs correspondientes a símbolos terminales y al análisis de una oración.

Siguiendo las convenciones de Prolog, las constantes se consignan con cadenas que empiezan con una letra minúscula y las variables, con una letra mayúscula. Términos coincidentes en algunos de sus valores se consignan mediante variables compartidas [Shieber89, Gazdar89]. Por ejemplo, la anterior gramática podría representarse en DCG:

```
s(Arg0,Sem) --> sn(Arg0), vp(Arg0,Sem).
sn(arg0(Num,Per,Gen)) --> word(sn,Num,Per,Gen,_).
vp(arg0(Num,Per,_),sem(Sem)) --> word(vp,Num,Per,_,Sem).
word(vp,sing,3,_,run) --> [runs].
word(sn,sing,3,masc,_) --> [he].
word(sn,sing,3,fem,_) --> [she].
```

Los términos en las DCG difieren de los DAGs, de las estructuras de rasgos y de notaciones como PATR en dos aspectos importantes [Shieber89]:

- *En el orden*: en vez de identificar los valores por su asociación con un rasgo, los términos se identifican por el orden lineal que tienen en la estructura de términos.
- *En la aridad*: el número de elementos en una estructura de términos es significativo para la unificación.

Ambos aspectos de DCG pueden conllevar una menor legibilidad respecto a PATR. Otra desventaja de las estructuras de términos se deriva de no hacer mención expresa de los rasgos que identifican una clase de valores, ya que ello dificulta la interpretabilidad de las estructuras: el constructor de una gramática debe recordar en todo momento qué posiciones argumentales corresponden a qué funciones. Finalmente, puesto que la aridad es significativa, si se quiere especificar el valor de uno de los elementos de un término, hay que especificar también todos los demás elementos, por lo menos marcándolos como variables. Esto conduce a unas gramáticas llenas de variables diseminadas por todas las reglas [Shieber89].

Por otra parte DCG aúna una serie de importantes ventajas cuando el objetivo final es la implementación de un sistema de PLN. En primer lugar, ya que pueden ser

introducidas prácticamente de manera directa en código Prolog y, puesto que existen compiladores eficientes de este lenguaje, las gramáticas escritas en el formato de DCG pueden ser compiladas directamente y convertirse en analizadores eficientes. En segundo lugar, a causa de la estrecha relación existente entre DCG y Prolog, las gramáticas DCG se pueden integrar fácilmente con programas escritos en Prolog. Por lo tanto resulta sencillo integrar un sistema de lenguaje natural escrito en DCG en otro programa, sea una base de datos o un sistema experto, pongamos por caso. Finalmente, los problemas de ingeniería asociados con el orden lineal y con los requisitos de la aridad de la unificación de los términos no son terriblemente gravosos para gramáticas pequeñas o medias. Como conclusión, en palabras de Shieber [Shieber89]:

> *"... puede que DCG sea el formalismo adecuado para el desarrollo rápido de sistemas simples y eficientes de lenguaje natural."*

Por lo que respecta a la comparación de DCG con otros formalismos lógicos "clásicos" en la comunidad de PLN, como las *gramáticas de extraposición* (*Xtrapositional Grammars, XG*) y las *gramáticas de metamorfosis* (*Metamorphosis Grammars, MG*), la siguiente cita de Abrahmson, concluyente tras la revisión de estos formalismos, puede resumir la situación [Abrahmson 89]:

> *"The three formalisms (DCG, XG, MG) .../... are as we have said, equivalent in terms of computational power. Which one to choose for a particular application is largely a matter of personal taste"*

4.4 Grandes bases de conocimiento

Durante la pasada década, se ha desarrollado una serie de proyectos que tienen como objetivo la construcción de grandes bases de conocimiento, de propósito general, que permitan su utilización por sistemas de IA. El objetivo fundamental de estas bases de conocimiento es disminuir el esfuerzo de desarrollo de los sistemas de IA mediante la reutilización de la información ya almacenada en la base de conocimiento [Lenat90a].

Desde un punto de vista teórico, algunos estudios han señalado la necesidad de disponer de un "conocimiento enciclopédico del mundo" para comprender o procesar, los diálogos o fragmentos de texto que se encuentran inmersos en un determinado contexto (en el sentido más amplio de la palabra). En palabras de Gazdar [Gazdar89]:

> *"...a theoretical model of language understanding is not complete without a model of knowledge representation and retrieval, and we cannot construct a robust understanding computing without providing it with a encyclopedic knowledge of the world."*

Desde este punto de vista, disponer de una base de conocimiento de propósito general es necesario para cualquier sistema de PLN. Disponer de una base de conocimiento previamente desarrollada conlleva una importante disminución en el

esfuerzo de desarrollo del sistema, e incluso puede determinar el que el desarrollo del sistema resulte factible o no.

Aun cuando para aplicaciones concretas, en dominios concretos, no resulte necesario disponer de todo este conocimiento "enciclopédico", disponer previamente de él puede conllevar, en todos los casos, una importante disminución en el esfuerzo de desarrollo del sistema.

Vamos a considerar dos tipos de proyectos orientados al desarrollo de estas grandes bases de conocimiento. El primero, representado por el proyecto Cyc, se propone desarrollar una base de conocimiento de propósito general, con información de carácter "enciclopédico" [Lenat86, Lenat90a, Lenat90b, Guha94]. El segundo, representado por proyectos como WordNet, es más restrictivo, y se orienta al desarrollo de bases de conocimiento léxico, exclusivamente. Estas bases de conocimiento incluyendo información relativa a todos los términos existentes en una lengua, como puede ser por ejemplo el Inglés [Miller93]. Trataremos a continuación ambos sistemas por ser los más representativos de ambas aproximaciones. Prestaremos atención especial a WordNet, por constituir una de las bases del sistema Ares, que presentamos en el capítulo siguiente.

4.4.1 Cyc: una base de conocimientos de propósito general

Cyc [Lenat86, Lenat90a, Lenat90b, Guha94] es un proyecto de la MCC (Microelectronics & Computer Technology Corporation) que pretende construir en el transcurso de una década una base de conocimientos a escala real, con un contenido más o menos equivalente al conocimiento que pueda tener una persona de educación media, o el existente en una enciclopedia de sobremesa. Más en concreto, se detallan tres tareas fundamentales en el proyecto [Lenat90a]:

i) Desarrollar un lenguaje para la representación del conocimiento.
ii) Desarrollar un conjunto de procedimientos para la manipulación del conocimiento.
iii) Construir la base de conocimiento, utilizando en el lenguaje desarrollado en i) y accesible mediante ii).

Los aspectos i) y ii) han sido cubiertos mediante la definición de un lenguaje, CycL, que constituye una ampliación de la lógica de predicados de primer orden [Lenat90a]. En CycL se encuentra representada la base de conocimiento en sí misma, CycKB. La dimensión de CycKB, en el proyecto [Lenat90a], se marca entre 10.000 y 100.000 veces mayor que la de un sistema experto típico actual (en la misma referencia se estima la de estos últimos entre 100-1000 reglas). En el estado actual [Guha94], el conocimiento que se incluye en CycKB es, efectivamente, general: información relativa a *tópicos genéricos* (tiempo, espacio, intenciones...), *tópicos específicos* (gente, lugares...) y *scripts de interrelación* (entre gente, objetos tangibles, lugares...), entre otros tipos. Como ejemplo concreto, en CycKB se almacena información fisiológica que se divide en tres tópicos interrelacionados: anatomía, alimentos y tratamientos médicos. Acerca de anatomía contiene información relativa a órganos como el corazón o el hígado, y para cada uno de ellos,

su número, tamaño y posición en los animales, así como cuáles de ellos son críticos para la supervivencia [Guha94].

Cyc se presenta como potencialmente útil para una gran cantidad de sistemas de IA, pero sus desarrolladores señalan especialmente su relación con el PLN. Por una parte Cyc puede proporcionar conocimiento útil para sistemas de PLN. Por otra parte, Guha señala la importancia que se da en el proyecto, a más largo plazo, de incluir un módulo de PLN en Cyc que permita la actualización del conocimiento contenido mediante la inclusión automática de información existente en texto (se hace notar el problema existente en torno a la información introducida en Cyc en 1990 y su desfasamiento cuatro años más tarde) [Guha94].

Disponer de bases de conocimiento como Cyc puede resultar determinante en el desarrollo de sistemas de PLN. El principal problema que el proyecto presenta en la actualidad es precisamente su completitud: el plazo de diez años estimado en principio [Lenat86] parece no ser suficiente. En efecto, de los tres aspectos centrales en el proyecto que señalamos al principio, los dos primeros parecen haberse completado satisfactoriamente con la definición de CycL, pero el desarrollo de CycKB, con las dimensiones marcadas en su inicio, parece no encontrarse aún próximo a su fin [Guha94].

4.4.2 WordNet: una base de conocimiento léxico

Con un objetivo más limitado que el de la aproximación representada por Cyc, se han desarrollado durante los últimos años una serie de proyectos centrados en la construcción de sistemas con información relativa al léxico completo de uno o varios idiomas. Como ejemplos de este tipo de proyectos podemos citar WordNet [Miller93], el léxico incorporado en el sistema ALVEY [Grover93] y el desarrollado en el proyecto ACQUILEX [Verdejo94], las dos primeras del Inglés y la última multilingüe.

WordNet es un sistema con información léxica extraída de forma semi-automática de diccionarios. Este sistema ha sido desarrollado en el Cognitive Science Laboratory de la Universidad de Princeton. Originalmente, en 1985, el desarrollo del sistema partió de un proyecto cuyo objetivo era producir un diccionario orientado a permitir búsquedas conceptuales en lugar de alfabéticas. Inicialmente, el sistema fue presentado por sus desarrolladores como un sistema inspirado en teorías psicolingüísticas relativas a la organización de la memoria léxica humana. En la actualidad, WordNet es un sistema totalmente desarrollado que incluye una base de datos de 12 MegaBytes en la versión 1.4, habiendo pervivido sólo las hipótesis teóricas más robustas [Miller93]. Una muestra de la información existente en WordNet se incluye en la figura 4.3.

En WordNet, el objeto básico es un conjunto de sinónimos estrictos denominado *synset*. Por definición, cada synset en que aparece una palabra representa un significado diferente de la palabra. Los synsets { board, plank } y { board, committee } pueden servir como designadores no ambiguos de los dos significados de "board". De forma intuitiva podemos hacer corresponder a un synset

```
Sense 1
car, gondola
  => compartment
Sense 2
car, elevator car
  => compartment
Sense 3
car, railway car, railroad car
  => wheeled vehicle
Sense 4
car, auto, automobile, machine,
motorcar, motor car
  => motor vehicle, automotive
     vehicle
```

a) synsets de "car"

```
Sense 4
car, auto, automobile, machine,
motorcar, motor car
 => motor vehicle, automotive
    vehicle
  => vehicle
   => conveyance, carrier,
      transport
    => instrumentality
     => artifact, article,
        artefact
      => object, inanimate
         object, physical
         object
       => entity
```

b) hipérnimos del synset #4

```
Sense 4
car, auto, automobile, machine,
motorcar, motor car
  HAS PART: accelerator, gun,
            throttle
  HAS PART: auto accessory
  HAS PART: automobile engine
  HAS PART: boot, luggage
            compartment, trunk
  HAS PART: buffer, fender
  HAS PART: bumper
  HAS PART: car door
  HAS PART: car mirror
  HAS PART: car window
  HAS PART: fender, wing, mudguard
  HAS PART: first gear, first, low
            gear, low

  ../..

  HAS PART: third gear, third
  HAS PART: window
  HAS PART: blinker, turn signal,
            turn indicator,
            trafficator
```

c) merónimos del synset #4

```
Sense 4
car, auto, automobile, machine,
motorcar, motor car
  => ambulance
  => beach wagon, station wagon,
     wagon, estate car
  => cab, hack, taxi, taxicab
  => bus, jalopy, heap
  => sports car, sport car
  => hot rod
  => compact, compact car
  => convertible
  => coupe
  => cruiser, patrol car, police
     car, prowl car, squad car
  => hardtop
  => hatchback
  => hearse
  => jeep, land-rover
  => limo, limousine
  => racer, race car, racing car
  => roadster, runabout, two-
     seater
  => sedan
  => stock car
  => touring car, phaeton, tourer
```

d) hipónimos del synset #4

Figura 4.3: Muestras de información asociada en WordNet a un término

con un concepto. Una palabra puede referenciar diferentes conceptos, o synsets. A su vez, un mismo concepto puede ser referenciado por varias palabras.

En WordNet también se almacena información relativa a las diferentes relaciones definidas entre palabras y synsets (o conceptos). Trataremos a continuación las más importantes: hiponimia, meronimia y antonimia, a parte de la misma sinomia, de la que se hace uso implícito en la definición de los synsets.

La *hiponimia* (*hyponymy*) es la relación existente entre conceptos o synsets equivalente a la relación *is-a*, o *es-un*. Un concepto representado por el synset { x, x', ... } es un hipónimo del concepto representado por el synset { y, y', .. } si un hablante del Inglés acepta oraciones construidas de la forma "an x is (a kind of) y". Por ejemplo, { maple } es un hipónimo de { tree } y { tree } es un hipónimo de

{ plant }. La relación de *meronimia* (*meronymy*) da cuenta de la inversa de *tiene-un* o *has-a*. Un concepto representado por el synset { x, x', ... } es un merónimo del concepto representado por el synset { y, y', .. } si un hablante del Inglés acepta oraciones construidas de la forma "a y has an x" , o "an x is a part of y". Por ejemplo, para el synset de "car" { car, auto, automobile, motorcar }, se encuentran como partes constituyentes synsets correspondientes a motor, ventana, puerta, acelerador, etc. (fig.4.3). A diferencia de la hiponimia y la meronimia, la relación de *antonimia* (*antonymy*) en WordNet se encuentra nivel de palabras, no de conceptos. Por ejemplo, los significados { rise, ascend } y { fall, descend } son conceptualmente diferentes, pero no son antónimos, sin embargo sí son antónimos los términos individualmente [rise / fall] y [ascend / descend].

La información de meronimia e hiponimia existente en WordNet es equivalente a la de las jerarquías definidas en redes semánticas [Frost89, Cortés93] mediante la relación es-un y tiene-un, con la importante característica de sus dimensiones: WordNet contiene actualmente 95.600 términos (51.500 correspondientes a palabras simples y 44.100 formados por más de una), organizados en 70.100 synsets.

En WordNet, se hace uso de la división del léxico en cuatro categorías: nombres, verbos, adjetivos y adverbios, de una forma análoga a los diccionarios comunes. De esta forma, los conceptos se encuentran agrupados en cuatro conjuntos disjuntos correspondientes a estas categorías. La mayor parte de la información almacenada corresponde a los nombres y los verbos. Para cada categoría se definen diferentes relaciones semánticas, además de las anteriormente citadas. Por otra parte, la meronimia sólo se incluye para los nombres, y para los adjetivos y adverbios no se define la hiponimia.

Desde el punto de vista del desarrollo de aplicaciones de PLN, las bases de conocimiento léxico permiten la reutilización de conocimiento sintáctico y semántico a nivel de léxico. Disponer de esta información para la construcción del léxico de un sistema de PLN puede conllevar una importante disminución en el esfuerzo de desarrollo, más aún teniendo presente la importancia creciente que ha adquirido el léxico en los sistemas de PLN durante la última década [Gazdar89].

4.5 Sistemas basados en el Procesamiento del Lenguaje Natural

Las aplicaciones en torno a las que se han desarrollado sistemas de PLN son múltiples: la traducción automática, los interfaces en lenguaje natural a bases de datos y a sistemas expertos, la corrección inteligente de textos y la generación automática de resúmenes son algunas de ellas [Verdejo94]. Un factor común de todas las aplicaciones es su potencial alta rentabilidad y su deseabilidad. Otra característica común es la dificultad de la implementación de los sistemas de este tipo. En general, existe un común acuerdo en reconocer la magnitud de los problemas que en múltiples vertientes presenta el desarrollo de sistemas de PLN [Gazdar89, Hobbs92, Boguraev95]. En este sentido, se considera excesiva la

tendencia existente en el campo al desarrollo de sistemas con una necesaria condición de prototipo, orientados a la investigación, e insuficiente la existencia de sistemas de dimensiones reales, resultando cuestionable su factibilidad en algunos casos [Hobbs92, Boguraev95].

Como ejemplo paradigmático de la evolución de la factibilidad de los sistemas de PLN, podemos considerar el caso de la traducción automática [Ballard87, Gazdar89]. La posibilidad de construir sistemas capaces de realizar la traducción automática de textos fue ya planteada durante los años cuarenta, apareciendo los primeros sistemas implementados con este propósito durante la década siguiente. El procesamiento que realizaban estos sistemas era extremadamente reducido, limitándose en la práctica a la sustitución de palabras mediante el uso de diccionarios bilingües. Este tipo de procesamiento no permitía la obtención de traducciones con unas mínimas características deseables. Después de década y media de investigación (importantemente financiada) en torno a estos sistemas, los avances conseguidos fueron mínimos y en el año 64 se formó un comité (ALPAC) para la evaluación del campo, en la que se concluía literalmente [Ballard87]:

> *"there has been no machine translation of general scientific text and none is in immediate prospect"*

El consiguiente cese de financiación motivó la práctica ausencia de investigación en torno a la cuestión durante más de una década. En contraste, la disponibilidad actual de sistemas comerciales, de uso frecuente en la traducción automática de manuales y diferentes tipos de documentación [Alonso94], indica un importante grado de factibilidad de una tarea de procesamiento que después de una inicial subestimación en su dificultad, parecía presentarse como irrealizable a medio plazo [Ballard87, Gazdar89]. Sin embargo, conviene señalar las limitaciones de los sistemas actuales: los valores típicos de la fiabilidad se encuentran entre el 60% y el 80% (lo que hace necesaria una revisión manual con la modificación del 20% al 40% del texto incorrecto) y ello resulta posible sólo para dominios determinados (manuales técnicos con control de la terminología empleada) [Alonso94].

En los siguientes párrafos se tratan sistemas de PLN que realizan labores que se encuentran especialmente relacionadas con las de los sistemas de recuperación de información, que se enfrentan al procesamiento de textos y que permiten a los usuarios formular solicitudes en lenguaje natural (LN). Se presta especial atención a la relación existente entre la definición del dominio y la factibilidad de los sistemas.

De esta forma tratamos los *sistemas de extracción de información* y los *interfaces en lenguaje natural*, que se centran en el procesamiento de textos en LN y en el de diálogos en LN con el usuario, respectivamente. El desarrollo de los sistemas de extracción de información es cronológicamente más reciente [Lehnert91]. Tratamos también los denominados *sistemas de comprensión de texto* que constituyen un paso previo a los sistemas de extracción de información concebidos como tales [Lehnert87, Lehnert91]. Finalmente tratamos en este punto los principales aspectos de los sistemas de recuperación de información desde el punto de vista del PLN.

4.5.1 Interfaces en lenguaje natural

Los interfaces en LN fueron una de las áreas de aplicación del PLN en torno a la que se desarrollaron los primeros prototipos de sistemas. Un ejemplo de estos primeros sistemas es BASEBALL. El sistema admitía consultas en LN referentes a información acerca de fechas y lugares de partidos celebrados en la liga de Baseball americana almacenados en una base de datos de dimensiones reducidas. Un ejemplo de entrada del sistema era [Ballard87]:

```
> What teams won 10 games in July?
```

Posteriormente ha aparecido una serie de sistemas orientados a permitir la interacción en LN con diversos tipos de sistemas informáticos, tales como sistemas operativos, sistemas expertos, sistemas de ayuda, sistemas de enseñanza asistida por computadora, etc. [Ballard87, Bates87, Wenger87, Wilensky89, Binot91, Li92, Iglesias93, Verdejo94, Ritchie95].

En la construcción de interfaces en LN se afrontan diferentes problemas lingüísticos que surgen al procesar las solicitudes de los usuarios. Citamos a continuación tres ejemplos, considerando interfaces en LN a bases de datos [Ritchie95].

Como primer problema consideramos el de la *asociación de modificadores*. Este problema se puede hacer patente en el procesamiento de consultas como la siguiente:

```
> List all employees in the company with a driving licence.
```

en la que se pueden considerar dos posibles asociaciones entre las que el sistema debe determinar la interpretación adecuada:

```
LIST ALL ([EMPLOYEES[IN THE COMPANY]][WITH A DRIVING LICENCE]).
LIST ALL [EMPLOYEES[IN THE[COMPANY[WITH A DRIVING LICENCE]]]].
```

Otro problema es el de la determinación del *ámbito de los cuantificadores*. Este problema se plantea al realizar la traducción de determinantes como "a", "each", "all" y "some" a cuantificadores lógicos. Por ejemplo, la consulta

```
> Has every student taken some course?
```

tiene dos posibles interpretaciones diferentes, correspondientes a comprobar:

$(\forall Student)\ (\exists Course)\ taken(Student, Course)$
$(\exists Course)\ (\forall Student)\ taken(Student, Course)$

Como último problema consideremos el de la *anáfora*, o la ambigüedad introducida por la utilización de pronombres (e.g. "she", "they"), determinantes posesivos (e.g. "his", "their"), y frases nominales para referenciar entidades previamente mencionadas. Este problema se hace patente en el siguiente diálogo:

```
> Who leads the TPI project?
E. Feron

> Who reports to him?
C. Leonard, C. Willems, E. Bidonnet, P. Cayphas, J.P. Van Loo

> What do they work on?
project      worker
------------------------------
DOCDIS       C. Willems
             J.P. Van Loo
             P. Cayphas
EURS         C. Leonard
             C. Willems
             E. Bidonnet

> Which of these are leaders?
J.P. Van Loo
```

Las ventajas de contar con una interfaz en LN radican en su flexibilidad y expresividad (i.e. el LN permite la expresión de cualquier tipo de información), así como en la accesibilidad que le ofrecen al usuario no experto u ocasional, de un sistema informático concreto, liberándole de aprender un lenguaje formal o unos procedimientos de consulta específicos. Los inconvenientes vienen ligados a las posibles restricciones de la interfaz en cuanto a vocabulario, formas de expresión, tolerancia de errores, o tipo de preguntas admitidas. Estas restricciones pueden imponer una forma de interacción que choque con las expectativas de "naturalidad" del usuario [Bates87, Verdejo94, Ritchie95]. En este sentido, el trabajo desarrollado en el campo se ha orientado a dotar a los sistemas de mayores capacidades de procesamiento con vistas a dar cuenta de los diferentes fenómenos que, como los anteriormente señalados, se presentan en los diálogos.

La adecuación del lenguaje natural como medio de interacción es una cuestión determinante al plantearse el desarrollo de un interfaz en LN. Si el sistema tiene una orientación que va más allá de la de mero sistema académico o experimental, pueden existir otros tipos de lenguajes más apropiados y que además supongan la implementación de un interfaz que conlleve un menor esfuerzo de desarrollo. Se pueden señalar una serie de circunstancias en las que un interfaz en LN no es el más adecuado [Bates87]. Por ejemplo, el LN no resulta adecuado como alternativa a controles manuales en sistemas como los simuladores de vuelo. Tampoco representa una alternativa adecuada a la interacción basada en iconos, menús desplegables y dispositivos señalizadores, como el ratón, en sistemas de diseño gráfico y tratamiento de imágenes. El desarrollo de lenguajes como SQL orientados a su utilización por usuarios inexpertos en el caso de las bases de datos y, en general, la evolución experimentada en los últimos años en el campo de la *interacción hombre-máquina* (*Human Computer Interaction, HCI*) [Maddix90], ha conllevado que para un cierto número de sistemas para los que un interfaz en lenguaje natural se presentaba como el más adecuado, lo haya dejado de ser.

El lenguaje natural resulta adecuado cuando el usuario no conoce las capacidades o limitaciones del sistema, cuando la tarea a realizar no se encuentra bien especificada, o cuando el esfuerzo de aprendizaje de un lenguaje formal o icónico no

se encuentra justificado [Bates87, Maddix90]. Un criterio general para fijar la adecuación del lenguaje natural se puede basar en la relación entre el número de funcionalidades del sistema utilizadas por el usuario y su tiempo de utilización [Bates87]. Los interfaces multimodales [Pérez94] proponen la inclusión de diferentes modos de interacción entre el usuario y el sistema: lenguaje basado en gráficos, en menús, LN, etc. Este tipo de sistemas permiten que la interacción entre el usuario y el sistema se realice mediante el lenguaje más adecuado. El desarrollo de un modelo de diálogo que facilite la integración de los diferentes modos de interacción es uno de los principales problemas que plantean este tipo de sistemas.

Por lo que respecta a la factibilidad de los sistemas, se han desarrollado prototipos orientados al tratamiento de problemas lingüísticos adicionales a los citados anteriormente, tales como la elipsis, la inferencia de objetivos (*goals*), restricciones de tipo social y expresiones extragramaticales que incluyen el tratamiento del contexto lingüístico, de las prestaciones del sistema informático con el que interactúa, la clase de usuarios a los que se dirige, el tipo de tarea que se lleva a cabo en la aplicación, etc. [Ballard87, Bates87, Carbonnell87, Noor92, Verdejo94, Allen94, Ritchie95]. Incluir todo este tipo de características puede dotar de una mayor "naturalidad" al interfaz, pero como contrapartida, conlleva un importante incremento en el esfuerzo de desarrollo, y su factibilidad, en algunas ocasiones, puede resultar cuestionable [Noor92]. En la actualidad, un número importante de los sistemas se encuentran orientados a la ejemplificación de modelos teóricos centrados en la resolución de problemas como los mencionados anteriormente y su condición de prototipo los hace poco útiles en la práctica [Bates87, Noor92, Ritchie95]. Por otra parte, también han surgido un cierto número de sistemas comerciales. Ejemplos de interfaces en LN a bases de datos de este tipo son INTELLECT, Q&A, RAMISII, NaturalLink o NLQ [Noor92]. Sin embargo, las capacidades de comprensión que presentan estos últimos sistemas y la rigidez impuesta a los usuarios en sus expresiones, hacen que no sean una alternativa competitiva a la interacción con los sistemas mediante otros tipos de lenguaje como los señalados anteriormente (e.g. icónico, formal) [Maddix90, Noor92, Ritchie95].

4.5.2 Sistemas de comprensión de texto

Una serie de sistemas desarrollados durante la década pasada tenían como objetivo la *comprensión de texto* o la *comprensión de historias* (*story understanding*) [Lehnert87]. Ejemplos concretos de tales sistemas son SAM [Cullingford81], PAM [Wilensky81], BORIS [Dyer83], FRUMP [Lehnert87] y DMAP [Martin89]. Si bien la condición de prototipo se encuentra presente en casi todos estos sistemas, su aportación resulta de interés especial en nuestro estudio por constituir la base de los sistemas de extracción de información y contribuir a la definición y aplicación de conceptos como el de *script* en el procesamiento de textos en LN (a parte por supuesto, de su contribución más general a la cuestión de la representación del conocimiento en Inteligencia Artificial [Frost89, Rich91, Cortés93]).

Estos sistemas se presentan en la bibliografía como capaces de realizar la comprensión del texto de entrada. Una serie de comparaciones o tests se realizan

Texto de entrada:

"Friday evening a car swered off Route 69. The vehicle struck a tree. The passenger, a New Jersey man, was killed. David Hall, 27, was pronounced dead at the scene by Dr. Dana Blauchard, medical examiner. Frank Miller, 32, of 593 Foxon Rd., the driver, was taken to Milford Hospital by Flanagan Ambulance. He was treated and released. No charges were made. Patrolman Robert Onofrio investigated the accident."

Resumen en Inglés:

AN AUTOMOBILE HIT A TREE NEAR HIGHWAY 69 FOUR DAYS AGO. DAVID HALL, AGE 27, RESIDENCE IN NEW JERSEY, THE PASSENGER, DIED. FRANK MILLER, AGE 32, RESIDENCE AT 593 FOXON ROAD IN NEW HAVEN, CONNECTICUT, THE DRIVER, WAS SLIGHTLY INJURED. THE POLICE DEPARTMENT DID NOT FILE CHARGES.

Resumen en Español:

UN AUTO CHOCO CONTRA UN ARBOL CERCA DE LA CARRETERA 69 EL VIERNES AL ANOCHECER. DAVID HALL, DE 27 AÑOS, QUE VIVIA EN NEW JERSEY, EL PASAJERO, SE MURIO. FRANK MILLER, DE 32 AÑOS, QUE VIVIA EN 593 CALLE FOXON EN NEW HEAVEN, CONNECTICUT, EL CHOFER, QUEDO UN POCO HERIDO. EL DEPARTAMENTO DE POLICIA NO ACUSO A NADIE.

Sesión preguntas-respuestas:

Q1: Was anyone killed?
A1: YES, DAVID HALL DIED.
Q2: Was anyone hurt?
A2: YES, FRANK MILLER WAS SLIGHTLY INJURED.
Q3: Why was Miller hurt?
A3: BECAUSE THE AUTOMOBILE HIT THE TREE.
Q4: Did Miller go to the hospital?
A4: YES, HE WAS IN THE MILFORD HOSPITAL
Q5: How did Miller get to the hospital?
A5: THE FLANAGAN AMBULANCE COMPANY TOOK HIM TO THE MILFORD HOSPITAL.

Figura 4.4: Ejemplo de entrada/salida del sistema SAM

para la comprobación de que la comprensión se ha realizado. En concreto se utilizan tres tipos de pruebas básicas [Lehnert87]:

- Respuesta a preguntas formuladas en torno al contenido del texto.
- Parafraseado del texto.
- Resumen del contenido del texto.

El sistema SAM (Script Applier Mechanism) [Cullingford81] es uno de los primeros sistemas basado en scripts. SAM utiliza scripts como información principal para la comprensión de textos relativos a accidentes de tráfico. El procesamiento de este tipo de textos resulta posible gracias a su estereotipificación, o estructura fija que presentan, y que es representada en los scripts. El sistema es capaz de generar un resumen del texto en Inglés, así como en un idioma diferente (Español en concreto) y responder a una serie de preguntas formuladas en LN. En la figura 4.4 aparece un ejemplo de la entrada/salida del sistema.

El sistema BORIS [Dyer83] presenta funcionalidades análogas, pero incorpora un mayor número de estructuras de conocimiento orientadas a la representación de información contextual además de los scripts, tales como *planes* (*plans*), *objetivos* (*goals*) relaciones entre personas, papeles sociales, reacciones emocionales, etc. Este gran número de tipos diferentes de información permiten al sistema comprender textos en los que las relaciones existentes entre los conceptos que aparecen son más complicadas. Por ejemplo, en el fragmento [Dyer83]:

> "When Paul walked into the bedroom and found Sarah with another man he nearly had a heart attack..."

BORIS es capaz de entender que "ataque al corazón" hace referencia a "gran sorpresa" dentro del papel de "marido engañado" (Sarah aparece anteriormente en el texto como la mujer de Paul) en el script de "adulterio" ("grosso modo", pues BORIS dispone de información más jerarquizada).

El sistema DMAP [Martin89] representa una evolución posterior de este tipo de ideas, incluyéndose además mecanismos que facilitan la gestión de la información (*organización de la memoria* utilizando su terminología) como los MOPS (*Memory Organization Packages*), enmarcados en el *Razonamiento Basado en Casos* (*Case Based Reasoning, CBR*) [Riesbeck89]. El enfoque adoptado en el desarrollo de estos sistemas se encuentra orientado a la ejemplificación de la utilización de diferentes estructuras de memoria en el proceso de comprensión de los textos. De acuerdo con esto, el dominio, o el corpus de textos que procesan, es muy reducido, y muchas veces estos textos son seleccionados por los autores de los sistemas con el único propósito de ejemplificar su funcionamiento. Para la orientación de nuestro trabajo, el principal problema que presentan estos sistemas es precisamente su factibilidad, o la posibilidad de desarrollar sistemas capaces de procesar corpus de textos, o dominios, de dimensiones reales.

Un sistema de comprensión de texto que se puede señalar como precursor de los sistemas de extracción de información [Hobbs92] es el sistema FRUMP [Lehnert87]. FRUMP se basa en la utilización de scripts, estructuras menos sofisticadas que las utilizadas en DMAP o BORIS. Sin embargo, resulta interesante la definición del dominio "a escala real" de textos a procesar que se plantea desde la misma concepción del sistema. FRUMP se basa en la utilización de scripts para la interpretación de noticias en formato electrónico. Dispone de unos 50 scripts diferentes, correspondientes a terremotos, secuestros, visitas diplomáticas y huelgas entre otros. En FRUMP interpretar una noticia es encontrar el script correspondiente e instanciar adecuadamente los valores de sus atributos, o lo que es lo mismo, obtener los valores correspondientes de los conceptos principales del texto concreto. El sistema representa la aproximación a la interpretación de texto denominada *text skimming* [Carbonell87]. En esta aproximación, en el procesamiento del texto se ignora un número considerable de las oraciones que aparecen en él, prestándose atención sólo a aquellas en las que se hace referencia a conceptos identificados como relevantes. Desde este punto de vista, guarda un importante paralelismo con los sistemas de extracción de información que tratamos a continuación.

4.5.3 Sistemas de extracción de información

El propósito de los sistemas de extracción de información es transformar la información almacenada en forma textual (e.g. libros, revistas e informes técnicos) a otra forma mucho más estructurada (por ejemplo, registros de bases de datos) [Jacobs90, Lehnert91, Hobbs92, Sandoval93, Verdejo94]. Los sistemas de extracción de información pueden considerarse como una evolución de los sistemas de

Texto de entrada:

TST1-MUC3-00
BOGOTA, 3 APR 90 (INRAVISION TELEVISION CADENA 1) - [REPORT] [JORGE ALONSO SIERRA VALENCIA] [TEXT] LIBERAL SENATOR FEDERICO ESTRADA VELEZ WAS KIDNAPPED ON 3 APRIL AT THE CORNER OF 60TH AND 48TH STREETS IN WESTERN MEDELLIN. ONLY 100 METERS FROM A METROPOLITAN POLICE CAI [IMMEDIATE ATTENTION CENTER]. THE ANTIOQUIA DEPARTMENT LIBERAL PARTY LEADER HAD LEFT HIS HOUSE WITHOUT ANY BODYGUARDS ONLY MINUTES EARLIER. AS HE WAITED FOR THE TRAFFIC LIGHT TO CHANGE, THREE HEAVILY ARMED MEN FORCED HIM TO GET OUT OF HIS CAR AND GET INTO A BLUE RENAULT. HOURS LATER, THROUGH ANONYMOUS TELEPHONE CALLS TO THE METROPOLITAN POLICE AND TO THE MEDIA, THE EXTRADITABLES CLAIMED RESPONSIBILITYOR THE KIDNAPPING. IN THE CALLS, THEY ANNOUNCED THAT THEY WILL RELEASE THE SENATOR WITH A NEW MESSAGE FOR THE NATIONAL GOVERNMENT. LAST WEEK, FEDERICO ESTRADA VELEZ HAD REJECTED TALKS BETWEEN THE GOVERNMENT AND THE DRUG TRAFFICKERS.

Registro completo:

0. MSG ID	TST1-MUC3-0080
1. TEMPLATE ID	1
2. INCIDENT DATE	03 APR 90
3. INCIDENT TYPE	KIDNAPPING
4. INCIDENT CATEGORY	TERRORIST ACT
5. INDIV PERPETRATORS	"THREE HEAVILY ARMED MEN"
6. ORG PERPETRATORS	"THE EXTRADITABLES" / "EXTRADITABLES"
7. PERP CONFIDENCE	REPORTED AS FACT:"THREE HEAVILY ARMED MEN" CLAIMED ADMITTED: "THE EXTRADITABLES" / "EXTRADITABLES"
8. PHYS TRAGET ID	*
9. PHYS TRAGET NUM	*
10. PHYS TRAGET TYPE	*
11. HUM TARGET ID	"FEDERICO ESTRADA VELEZ" ("LIBERAL SENATOR"/"ANTIOQUIA DEPARTMENT LIBERAL PARTY LEADER" / "SENATOR" / "PARTY LEADER")
12. HUM TARGET NUM	1
13. HUM TARGET TYPE	GOVERNMENT OFFICIAL/POLITICAL FIGURE: "FEDERICO ESTRADA VELEZ"
14. FOREIGN TGT NAT'N	*
15. INSTRUMENT TYPE	*
16. INCIDENT LOCATION	COLOMBIA: MEDELLIN (CITY)
17. PHYS TGT EFFECT	*
18. HUM TGT EFFECT	*

Figura 4.5: Ejemplo de entrada/salida de un sistema en MUC-3

comprensión de texto [Hobbs92]. A diferencia de los sistemas de comprensión de texto, en la concepción de los sistemas de extracción de información se parte de la definición precisa de la funcionalidad del sistema y de la definición del dominio de los textos a procesar.

En 1991 se celebró la tercera conferencia en comprensión de mensajes, MUC-3 (Message Understanding Conference), dentro del programa Tipster de ARPA [Lehnert91, Harman92, Boguraev95]. Las dos primeras se realizaron en el 87 y el 89, respectivamente. El objetivo de la conferencia era la evaluación del estado del arte de los sistemas y técnicas de extracción de información. En MUC-3 participaron 15 sistemas, tres desarrollados en universidades y doce en instituciones privadas. El propósito de la conferencia era doble: poder comparar el resultado de los sistemas frente a una misma tarea, y establecer criterios y métodos de evaluación que permitan la comparación de posibles sistema futuro. La tarea que debían realizar los sistemas era procesar una serie de noticias relativas a acciones terroristas, asignando los valores adecuados a un registro o plantilla previamente definida. En figura 4.5 aparece un ejemplo de texto a procesar y el resultado obtenido [Lehnert91].

Una cuestión importante en el planteamiento de MUC-3 fue la definición precisa de la funcionalidad que debían implementar los sistemas y la del dominio de los textos a procesar. El partir de la definición de ambas cuestiones representa una evolución importante respecto a la metodología de desarrollo seguida en los sistemas de comprensión de textos. En este último tipo de sistemas la definición de la funcionalidad y del dominio juegan un papel secundario en el desarrollo de los sistemas frente al tipo de técnica de análisis (e.g. scripts) utilizada, que juega un papel prioritario.

En el corpus desarrollado se utilizaron 1300 textos procedentes de noticias de periódicos y revistas. En el desarrollo del corpus para la evaluación se incluyó además el análisis de los textos, realizado manualmente, para la comparación con los obtenidos por los sistemas. Como datos concretos, en el corpus se incluyeron como incidentes más numerosos 404 correspondientes a asesinatos, 270 correspondientes a colocación de bombas y 92 a secuestros. El corpus completo contiene aproximadamente 400.000 palabras compuestas de 18.240 ítems léxicos únicos, 15.600 oraciones con una longitud media de 27 palabras. Cada texto del corpus contiene una media de 12 oraciones [Lehnert91].

Conviene subrayar la importancia otorgada en proyecto al desarrollo del corpus, el esfuerzo destinado y la organización de repartos de tareas que lo hizo posible [Lehnert91]:

> *"To our knowledge, the distribution of labor behind the MUC-3 development corpus is unprecedented in AI (at least outside Japan) and was a critical component in making this corpus possible"*

Los sistemas desarrollados utilizaban técnicas de análisis de diferentes tipos, basadas en *ajuste de patrones* (*pattern matching*), análisis sintáctico y semántico, y técnicas estadísticas. En una convocatoria posterior, MUC-4, los resultados de los sistemas mejoraron y una serie de conclusiones se hicieron patentes, nos interesa subrayar [Verdejo94]:

- Todos los sistemas tuvieron que plantearse fases de pre o post proceso, donde los problemas no son de carácter lingüístico. Esta cuestión no se considera en los prototipos de investigación, pero es fundamental en aplicaciones reales y pone de manifiesto la necesidad de enfrentar con técnicas de Ingeniería del Software el desarrollo de sistemas de procesamiento de lenguaje natural

- El obstáculo más importante es el coste de transportar el sistema de una aplicación a otra (cambio de dominio y/o tarea), derivado del problema de la adquisición del conocimiento léxico y del dominio.

Otras cuestiones cuya investigación se señala como fundamental para el desarrollo de sistemas en este tipo de proyectos son [Evans93]:

- La utilización selectiva del PLN.
- La combinación de PLN con heurísticas y otras técnicas.
- Medios prácticos para esquivar problemas del PLN.
- Técnicas para utilizar información sintáctica con vistas a conseguir funcionalidad semántica.

Todo este esfuerzo investigador ha producido sistemas prácticos importantes, y en la actualidad, sistemas de extracción de información con propósitos comerciales (sobre diferentes dominios) se utilizan en compañías como Boeing, BBN, GTE y GE fundamentalmente para la inclusión de información en bases de datos a partir de mensajes telegráficos e informes técnicos [Hobbs92, Friedman95]. En estos sistemas se continúan encontrando presentes la especificidad del dominio y el esfuerzo de

desarrollo. Como ejemplo de los problemas relativos a cuestiones de preprocesamiento específicos del dominio que, como señalamos anteriormente, no se consideran en prototipos de investigación, podemos considerar el de identificación de los nombres propios de compañías comerciales (e.g. su nombre completo, su acrónimo, abreviatura...) en textos relativos a informes económicos [Jacobs90].

Finalmente conviene señalar, por lo que respecta a aspectos relacionados con la definición del dominio y el desarrollo de métodos de evaluación, cómo la evolución experimentada en los sistemas de extracción de información no se ha alcanzado en el desarrollo interfaces en LN hoy por hoy. Proyectos como los centrados en el procesamiento de textos (e.g. MUC, o TREC dentro del programa Tipster de ARPA) [Lehnert91, Harman92, Boguraev95] no tienen aún un equivalente en el campo de los interfaces en LN [Ritchie95].

4.5.4 Sistemas de recuperación de información

Sistemas de recuperación de información que tienen como entradas tanto consultas como textos en LN implementan una funcionalidad que incluye aspectos de los interfaces en LN y de los sistemas de extracción de información. En estos sistemas las expresiones en LN son traducidas a un lenguaje interno estructurado.

En los sistemas de RI actuales se tiende a explotar los avances experimentados en HCI [Maddix90, TMC91, Downton91, Sonnewald92], asignando a cada elemento de la interacción el lenguaje adecuado (aproximación seguida en la definición del sistema Argos). De esta forma, se utilizan lenguajes de manipulación directa [Maddix90], para una gran parte de la interacción (e.g. mediante elementos como menús, ventanas, botones y dispositivos señalizadores como el ratón). Por otra parte, el LN es utilizado en ellos para la especificación de forma declarativa de las necesidades del usuario, resultando el más adecuado para esta función frente a otros (e.g. formales o icónicos) [Cleverdon91, Sonnewald92]. Desde el punto de vista del tipo de expresiones utilizadas por los usuarios, la relevancia de determinados fenómenos lingüísticos puede ser diferente que en la interacción en LN con otros tipos de sistemas (e.g. el problema de determinación del ámbito de los cuantificadores en los interfaces en LN a bases de datos).

En los puntos siguientes analizaremos con mayor detalle los mecanismos utilizados para el análisis de los textos en los sistemas que integran técnicas de PLN en la RI. Esta aproximación se beneficia del hecho de que, como en los sistemas de extracción de información, la existencia previa de corpus de textos puede facilitar la definición del dominio y el desarrollo de los sistemas. Por otra parte, elementos utilizados para el análisis de los textos pueden ser utilizados para el procesamiento de las consultas si el lenguaje utilizado en ellas por los usuarios presenta semejanzas con el de los documentos, como ocurre en las consultas formuladas en sistemas de RI [Guindon88].

4.6 Sistemas de recuperación de información basados en la sintaxis

Métodos de Recuperación de Información basados en el uso de información sintáctica fueron ya ideados y evaluados en el proyecto SMART a principios de los años 70 [Salton83, Salton91a]. Los resultados obtenidos no fueron positivos. Se señalaba como principal inconveniente la falta de perfección de los métodos y herramientas de análisis lingüístico [Salton83]:

> *"The conclusion is that the role of linguistic methodologies in general and of syntactic analysis in particular is still unresolved for information retrieval. Before reaching a final conclusion in this area, it is wise to wait for the appearance of more sophisticated language analysis methods that are at the same time sufficiently efficient to permit incorporation into operational retrieval frameworks. Such methods should then be thoroughly evaluated to determine their actual value in information retrieval."*

Los "sophisticated language analysis methods" esperados se consiguieron en parte con los avances realizados en el PLN en los siguientes años, especialmente por lo que respecta a analizadores sintácticos, y como consecuencia aparecieron nuevas propuestas. En la bibliografía [Fagan87, Smeaton88, Metzler89, Salton89b, Croft91, Evans91], pueden encontrarse diferentes tipos de sistemas basados esencialmente en el análisis sintáctico de las oraciones de los documentos. Aunque existen diferencias entre cada uno de los sistemas, se puede señalar una idea básica como fundamento común de todos ellos [Lewis92, Hess92]: obtener para cada oración una o más tuplas de términos (normalmente parejas), después de aplicar un procedimiento de normalización a las estructuras sintácticas obtenidas, basadas en determinadas relaciones gramaticales entre ellos. Por ejemplo, se puede obtener una tupla de términos a partir del verbo y el núcleo de su sujeto, o a partir de un nombre y un adjetivo que le modifica. El objetivo es, por lo tanto, el de la construcción de frases de términos, utilizándose criterios sintácticos adicionales a los basados en criterios de frecuencias de aparición de términos, tratados en capítulos anteriores. La indexación basada en estas frases de términos tiene como finalidad principal la representación con más precisión de los documentos.

La superficialidad (shallow) del análisis sintáctico realizado varía de unos sistemas a otros. En algunos, el análisis se basa en un conjunto de patrones relativamente sencillos como <ADJ-NN> (adjetivo nombre) o <NN-PP-NN> (nombre preposición nombre), mientras que en otros el análisis sintáctico que se realiza incluye un mayor detalle.

Desde el punto de vista de las gramáticas de unificación, las reglas gramaticales utilizadas en los analizadores de los sistemas enmarcados en esta aproximación son específicas del idioma de los documentos que se procesan (e.g. Inglés), pero no de un dominio concreto, en el sentido de los sistemas de comprensión de texto o de extracción de información. Este hecho marca una diferencia importante entre los

sistemas que siguen esta aproximación y los basados en la aproximación semántica, que tratamos en un punto siguiente.

En la bibliografía, los sistemas basados en la sintáxis se suelen presentar como *syntactic indexing*, en tanto en cuanto se pone el énfasis en la obtención de frases de términos mediante los que representar, o indexar, los documentos. El sistema de Fagan [Fagan87] constituye, para nuestro estudio, el mejor representante de esta aproximación. Se basa en el análisis sintáctico de las oraciones para la obtención de frases de términos. La aproximación de Fagan puede resumirse como sigue:

1. Utilizando un analizador sintáctico, producir un árbol de análisis para una oración.

2. Seleccionar del árbol de análisis todos los pares de palabras que satisfacen configuraciones estructurales específicas. Normalmente, esta configuraciones se eligen de forma tal que una palabra es el núcleo de un sintagma nominal y el otro un modificador, aunque se procesan también algunas construcciones verbales.

3. Agrupar en una sola frase de términos todas las parejas núcleo-modificador basadas en palabras equivalentes tras su extracción de raíces. Por ejemplo, se obtiene "quer analys" para las siguientes oraciones:

   "They designed a query analysis system"
   "They designed a system for analyzing the queries"
   "The system analyzing the queries is automatic"
   "The queries analyzed by the system are well-constructed"
   "They designed a system to analyze queries automatically"

4. Indexar los documentos por aquellas frases de términos cuya frecuencia de aparición no sea demasiado alta.

Fagan utiliza un analizador sintáctico previamente disponible, el sistema PLNLP, para analizar completamente el texto de dos colecciones de documentos de prueba y obtener listas de frases de términos como índices. La sofisticada gramática incluida en el sistema PLNLP le permite manejar sintagmas nominales (noun phrases) complicados con modificadores precedidos por preposiciones y una gran cantidad de construcciones diferentes. El sistema PLNLP proporciona varios análisis para cada oración ordenados de forma decreciente por su corrección estimada, de los que se utilizan los primeros para la construcción de las frases de términos. También se incluye en el sistema una lista de palabras especiales que indican cuando generar o no generar un índice, o generarlo de una forma especial.

Curiosamente, el incremento en la efectividad conseguido mediante esta aproximación no es especialmente significativo. Fagan utiliza dos colecciones de documentos y consultas (CACM y CISI) para el desarrollo de una serie de experimentos en los que compara la indexación basada en frases de términos obtenidas sintácticamente con frases de términos obtenidas mediante criterios estadísticos. La efectividad que consigue con las frases de términos obtenidas por

métodos sintácticos es diferente para cada colección: consigue incrementos del 1.2% y 8.7% en la precisión respecto al indexado por términos simples, respectivamente. Por otra parte, estas mejoras son inferiores a las obtenidas mediante métodos estadísticos de generación de frases de términos: incrementos del 2.2% y 22.7%, respectivamente.

Algunos sistemas como CLARIT [Evans91] y COP [Metzler89], se presentan en la bibliografía como una aproximación diferente a la anterior. El sistema COP (Constituent Object Parser) se presenta como basado en "syntactic structure matching" en lugar de "syntactic indexing". COP no realiza un análisis completo de los documentos para obtener sus índices, sino que, esencialmente, compara la oración de la consulta con las oraciones de los documentos. Sin embargo, aunque la secuencia en que se realicen los análisis sea diferente, la idea básica antes citada como subyacente a todos, la representación por frases de términos, sigue estando presente [Hess92].

En general, las mejoras conseguidas en la efectividad del proceso de recuperación mediante la utilización de técnicas basadas en la sintáxis son menores que las conseguidas por los sistemas basados en la semántica, que tratamos en el punto siguiente. Por otra parte conviene señalar que en trabajos orientados a la investigación de otras operaciones sobre términos y documentos (e.g. generación de listas de asociación de términos, categorización de textos) se han conseguido mejoras significativas mediante la introducción de criterios basados en el análisis sintáctico [Lewis92, Grefenstette92].

4.7 Sistemas de recuperación de información basados en la semántica

Cualquier sistema capaz de comparar la solicitud del usuario con el contenido de los documentos basándose en la semántica, o su significado, se puede presentar como un candidato plausible a la mejora de la efectividad de la Recuperación de Información [Mauldin91, Salton89b, Hess92. En la bibliografía [Tong87, Jacobs88, Wendlandt91, Mauldin91, Rau91, Hess92, Kupiec93, Voorhees93] se pueden encontrar una serie de sistemas que se presentan como sistemas de Recuperación de Información "basados en la semántica", o "conceptuales". Estos sistemas se encontrarían incluidos en los que trataremos en este punto como aproximación basada en la semántica. Es decir, la aproximación seguida por aquellos sistemas que incluyen conocimiento, o información, adicional a la sintáctica. Sistemas de este tipo han aparecido durante los últimos años de forma simultánea al avance experimentado por los sistemas de PLN. Estos sistemas realizan el análisis lingüístico a niveles superiores al sintáctico, y son capaces de procesar corpus de textos de dimensiones reales, de una forma análoga a los sistemas de extracción de información.

Las mejoras en la efectividad obtenidas por los sistemas basados en la semántica son en general mayores que las conseguidas mediante los sistemas basados en la sintáxis. El principal problema de estos sistemas es precisamente el mayor coste de

Article 1307 of 1351, Mar 3 02:00.
Subject: Pioneer 10
From: dipper@utastro
Organization: U. Texas, Astronomy, Austin, TX
Newsgroups: net.astro
Date: Mon, 3-Mar-86 02:00:18 EST
Pioneer 10 was the first spacecraft to venture into the outer solar system. More -- coming up.
March3 PioneerIO
On today's date in the year 1972, NASA's Pioneer 10 spacecraft was launched toward the outer solar system.
It was to become the first craft to travel beyond the asteroid belt -- and the first to encounter mighty Jupiter...

Article 192 of 4 Aug 85.
Subject: Jupiter at Opposition
From: dipper@utastro
Organization: U. Texas, Astronomy, Austin, TX
Newsgroups: net.astro
Date: 4-Aug-85 02:00:00 EST
The planet Jupiter today falls behind Earth in the race around the sun. More -- after this.
August4 JupiteratOpposition
You can probably spot Jupiter now, and for the next few months, as the very bright object in the east each evening.
Jupiter shines brillianlly by virtue of its large size, its bright cloud cover which reflects sunlight so well, and its relative nearness to Earth right now...

Figura 4.6: Fragmentos de textos procesados por FERRET.

desarrollo que conllevan respecto a los sistemas basados en la sintaxis. De hecho, algunas propuestas que cabrían incluirse aquí, como la del sistema LogDoc [Hess92], no son más que breves esbozos de sistemas, si bien otros han sido completamente desarrollados. Además, los sistemas basados en la semántica, a diferencia de los basados en la sintaxis, trabajan sobre documentos de dominios específicos, por lo que su utilización en un dominio diferente del original conlleva en la práctica la implementación de un nuevo sistema, con el consiguiente esfuerzo de desarrollo.

En esta aproximación nos interesa diferenciar dos tipos de sistemas: sistemas basados en la interpretación del texto y sistemas basados en el conocimiento léxico. Los sistemas basados en la interpretación del texto realizan un análisis lingüístico más profundo de los textos, próximo al de sistemas de PLN como los de extracción de información o los de comprensión de textos. Sistemas de este tipo que trataremos a continuación con mayor detalle son FERRET [Mauldin91] y NLDB [Rau91]. Los sistemas basados en el conocimiento léxico realizan un análisis más superficial que los anteriores, utilizando información casi exclusivamente a nivel de léxico, sin utilizar prácticamente ningún tipo de reglas gramaticales. Sistemas de este tipo que trataremos son UCF [Wendlandt91] y el sistema de Voorhees [Vorhees93].

4.7.1 FERRET

Mauldin desarrolla el sistema FERRET al que presenta como un "Conceptual Information Retrieval System" [Mauldin91]. El sistema opera sobre un dominio concreto: el definido por 1065 textos relacionados con eventos astronómicos obtenidos del Observatorio McDonalds de la Universidad de Texas. En la figura 4.6 se incluyen dos ejemplos de los documentos procesados. La longitud media de los textos es de 300 palabras y unos 800 caracteres. Para el procesamiento de los documentos, FERRET utiliza como información principal una base de scripts y un léxico formado por palabras con significados asociados. Utiliza como analizador McFRUMP, una versión simplificada del analizador utilizado en el sistema FRUMP

[Lehnert87], un analizador conceptual ya tratado en un punto anterior de este capítulo. La base de conocimientos utilizada en FERRET está formada por:

- 5 scripts básicos
- 965 frames
- 442 palabras
- 272 significados de palabras
- 251 conceptos

FERRET, de forma análoga a los sistemas de PLN basados en scripts, obtiene como resultado del análisis de un texto una instanciación del script correspondiente. Ya que el proyecto FERRET se centra en el procesamiento de los textos más que en el de las consultas, se utiliza para el análisis de estas últimas el mismo analizador de los documentos con algunas modificaciones, a modo de prototipo. De esta forma, se obtienen también scripts instanciados parcialmente como resultado del análisis de las consultas. El cálculo de la similitud entre los scripts instanciados se basa en un procedimiento de ajuste de patrones (pattern-matching).

Los resultados obtenidos por el sistema son comparados con otro sistema basado en palabras clave. Utilizando un conjunto de 22 preguntas y 532 documentos, obtiene valores de la Precision del 48% para FERRET frente a un 35% para el basado en palabras clave. El Recall se ve mejorado aún más, obteniendo 52.4% para FERRET frente a un 19.4 % para el basado en palabras clave.

De acuerdo con lo previamente expuesto, FERRET representa la factibilidad del uso de sistemas de PLN que realizan una comprensión parcial del texto para la RI. Interesa también notar el importante coste del desarrollo de este tipo de sistemas y la especificidad de su dominio. Si bien la aproximación seguida en FERRET puede trasladarse a dominios diferentes, esto implica en la práctica reconstruir la base de conocimientos.

4.7.2 NLDB

El sistema NLDB (Natural Language DataBase) [Rau91] ha sido desarrollado para procesar noticias económicas en formato electrónico. El número de documentos sobre el que trabaja es mucho mayor que FERRET, del orden de miles. De hecho, NLDB forma parte de un producto comercial de selección y recuperación de noticias y se encuentra inserto dentro de un entorno que incluye numerosas facilidades gráficas. El sistema ha sido implementado, por lo que al análisis de los textos respecta, de forma muy semejante a los sistemas de extracción de información, incluyendo una gran cantidad de heurísticas y conocimiento específicos del dominio: el reconocedor de nombres propios de empresas, por ejemplo, constituye una parte importante del sistema.

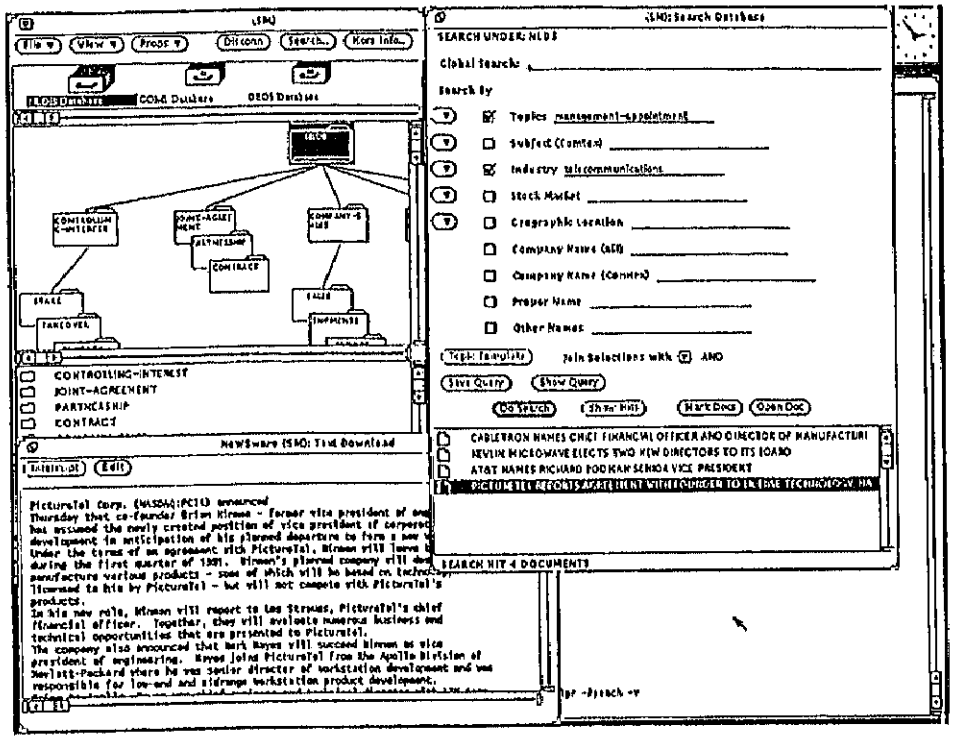


Figura 4.7: Interfaz del sistema NLDB

El resultado del análisis que NLDB realiza de los documentos es la instanciación, para cada documento, de un registro con campos como "tópico", "tipo de industria", "localización geográfica", etc. De esta forma se construye una base de datos relacional en la que cada documento está representado por uno de estos registros. El usuario rellena un registro con los valores de los campos adecuados para la recuperación de los documentos que mejor se adapten a su necesidad (fig. 4.7). Disponer de esta representación permite realizar búsquedas de los documentos que no son posibles mediante palabras clave: por ejemplo, podemos solicitar noticias relativas a un determinado tipo de industria.

Problemas como los relacionados con la polisemia y la sinomia son también resueltos en parte importante. Por ejemplo, si la palabra "prime" es incluida en la consulta en el campo "company name", NLDB proporciona documentos relativos a compañías como "prime computer", sin embargo, si se introduce en el campo "text-subject" se le asocia el significado de "prime rate"; NLDB dispone también de mecanismos que le permiten identificar diferentes nombres de una misma compañía como equivalentes. En cualquier caso, y de forma complementaria, en el sistema siempre se encuentra disponible una opción de búsqueda de documentos basada en frecuencia de aparición de palabras.

DOC 3976

TEXT:

The shuttle will transport cargo into near Earth orbit
100 to 217 nautical miles (115-250 statute miles)
above the Earth. This cargo (called payload) is
carrired in a bay 15 feet in diameter and 60 feet long.

BOTTOM OF DOCUMENT

Figura 4.8: Ejemplo de documento procesado por UCF.

4.7.3 UCF

En contraste con las aproximaciones seguidas en FERRET y NLDB, el análisis propuesto por Wendlant [Wendlant91] en el sistema de recuperación de documentos UCF, se basa casi exclusivamente en el uso de información semántica a nivel de léxico.

El sistema trabaja con el manual de mil páginas del transbordador espacial de la NASA, considerando como documento cada párrafo del manual (figura 4.8). De esta forma el conjunto de documentos resultante consta de 4902 elementos. Documentos y consultas se comparan utilizando una representación semántica basada en la gramática de casos de Fillmore: se utiliza una lista de veinte roles temáticos, algunos de ellos especialmente adecuados a su dominio (e.g. TIME, AMOUNT, LOCATION, CONDITION, COMPARISON).

La información semántica utilizada por UCF se encuentra fundamentalmente a nivel léxico. El sistema utiliza una lista de 32 preposiciones "indicadoras" (*triggers*) de uno o varios roles temáticos (figura 4.9). Además de esta lista de preposiciones indicadoras, dispone de otra análoga, de 34 palabras indicadoras con un único rol temático asociado, de forma en gran medida específica de este dominio. Como ejemplo podemos considerar las cinco primeras:

any	AMOUNT
approximately	AMOUNT
area	LOCATION
base	LOCATION
California	LOCATION

Utilizando esta información, se asignan pesos a los roles temáticos para cada oración dependiendo de las preposiciones y palabras que en ella aparecen. Wendlant presenta criterios basados en el número de roles asociados a cada preposición, y la frecuencia de aparición de las palabras y las preposiciones en los documentos, que le permiten calcular pesos de roles temáticos análogos a los de los términos. Así define el valor wr_{ij}, del peso de un rol temático r_i en un documento d_j. Esto le permite

```
after:    TIME, NONE
above:    AMOUNT, LOCATION, NONE
as:       CONDITION, COMPARISON, MANNER,
          NONE
at:       CAUSE, CONDITION, LOCATION,
          MANNER,
          RANGE, TIME
atop:     LOCATION
before:   LOCATION, TIME
below:    AMOUNT, LOCATION, NONE
between:  COMPARISON, DURATION, RANGE, NONE
by:       AMOUNT,CONVEYANCE,
          INSTRUMENT, LOCATION, TIME
during:   DURATION, TIME
except:   CONDITION, NONE
for:      BENEFICIARY, CAUSE, DURATION,
          GOAL
from:     CAUSE, RANGE, SOURCE, TIME
how:      AMOUNT, CONDITION, DEGREE,
          MANNER
in:       CONDITION, CONVEYANCE,
          DESTINATION, INSTRUMENT,
          LOCATION, MANNER, PURPOSE, TIME
into:     CONDITION,LOCATION, SPACE, TIME
like:     COMPARISON
of:       AMOUNT, CAUSE, LOCATION,
          PURPOSE, SOURCE, NONE
on:       CONDITION, CONVEYANCE,
          INSTRUMENT, LOCATION, SOURCE,
          TIME, NONE
over:     DURATION, INSTRUMENT, LOCATION,
          TIME, NONE
per       AMOUNT, TIME
through:  CONDITION, DURATION,
          INSTRUMENT, RANGE,
          RESULT, NONE
to:       ACCOMPANIMENT, BENEFICIARY,
          CONDITION, DEGREE, DESTINATION,
          LOCATION, PURPOSE,
          RANGE, RESULT, NONE
under:    CONDITION, LOCATION
until:    CONDITION, TIME
upon:     CONDITION, CONVEYANCE,
          INSTRUMENT,
          LOCATION, SOURCE, TIME, NONE
when:     TIME
where:    LOCATION
why:      CAUSE, PURPOSE
with:     ACCOMPANIMENT, BENEFICIARY,
          CAUSE, COMPARISON, CONDITION,
          INSTRUMENT, MANNER, RESULT
within:   CONDITION, RANGE
without:  CONDITION
```

Figura 4.9: preposiciones indicadoras de roles temáticos.

definir una función de similitud basada no en los pesos de los términos de los documentos y de la consulta, sino en los roles temáticos que intervienen en ellas.

Por ejemplo, para una consulta como:

```
How much does the space shuttle weight?
```

El valor del peso de un rol temático le permite identificar como atributo de mayor peso WEIGHT, en contraste con un peso pequeño para atributos como COLOR, SIZE, ORDER, que, por otra parte, son de frecuente aparición en los documentos, pero no en esta consulta.

Finalmente, el sistema también utiliza un conjunto de palabras a las que se les asocia una categoría semántica más general, de forma análoga a los thesauri. Por ejemplo:

```
abort          CHANGE
acceleration   GENERAL_MOTION
altitude       EXTERNAL/INTERNAL_DIMENSIONS
angle          LINEAR_DIMENSION
ascent         CHANGE_OF_PLACE
```

Esta información se incluye en el léxico del sistema para 128 palabras. El proceso de definición del léxico se realiza de forma manual, tomando como base para la definición de las clases de las palabras la existente en el Roget's Thesaurus [Edwards92]. Esta información relativa al significado de las palabras es también incluida en el proceso de indexación y cálculo de la similitud mediante criterios análogos a los anteriormente citados para los roles temáticos.

En UCF, el análisis semántico no se presenta como alternativa al basado en frecuencias de términos. Para recuperar los documentos dada una consulta, primero se utiliza un método de recuperación basado en el modelo vectorial, pesos de términos, extracción de raíces y lista de parada, y, después sobre el conjunto de documentos recuperados, se utiliza la función de similitud basada en los roles temáticos y las categorías semánticas para reordenar los documentos por su relevancia. En este sistema, por lo tanto, el análisis semántico se utiliza como un mecanismo mejorador de la Precision y complementario al de la búsqueda basada en términos.

Wendlandt utiliza una colección de 21 consultas para estudiar la efectividad de su sistema respecto a una aproximación basada en palabras clave, y obtiene mejoras significativas que oscilan entre el 36.6% y el 48.9% en parámetros relacionados esencialmente con la precisión. Interesa subrayar el coste de desarrollo de este sistema, pese a su aproximación "minimalista". Este alto coste es debido a la dificultad que conlleva la obtención manual de la información léxica utilizada, específica del dominio (formada esencialmente por el diccionario de preposiciones y términos con roles temáticos asociados, y el de términos con categorías semánticas asociadas):

> *"A dictionary and thesaurus were used to manually determine triggered thematic roles and attributes for each trigger. This was a tedious task and resulted in a semantic lexicon of 194 triggers..."*

A la cuestión del esfuerzo de desarrollo, se le suma en este sistema la de la escalabilidad. El léxico ha sido construido sólo para un subconjunto del total de los documentos del dominio, en concreto para 160 párrafos y las 21 consultas utilizadas en los experimentos. La definición de información léxica adicional resultaría necesaria para el procesamiento de todo el dominio considerado inicialmente.

4.7.4 Voorhees

Ellen M. Voorhees propone el uso de parte la información léxica existente en WordNet para la implementación de un sistema de RI utilizando adicionalmente criterios estadísticos [Voorhees93]. En su aproximación, intenta disminuir los problemas que plantean, para la RI, la polisemia y la sinonimia. Para ello utiliza principalmente la información existente en WordNet acerca de la relación *is-a* (hipernimia) para los nombres del Inglés. La información lingüística utilizada para el análisis es por lo tanto, al igual que en UCF, esencialmente semántica a nivel de léxico.

Define el *hood* de un significado o synset como un "concepto general" o "categoría de significados" asociado a él, algo análogo a las clases definidas en los thesaurus, pero adaptado a synsets. Para definir el hood de un synset s, considera el conjunto de synsets y los enlaces de hiponimia is-a de WordNet como el conjunto de vértices y arcos dirigidos de un grafo. El hood de s es el subgrafo conectado más grande que contiene s, contiene los descendientes directos de un ascendiente de s, y no contiene ningún synset que tenga un descendiente que sea también miembro del hood. El

hood es representado por el synset que se encuentra en la raíz del hood. Para una palabra, por lo tanto, tenemos varios hoods o "significados generales" asociados. Con la definición de hood, se intenta construir una especie de thesaurus más general que el que proporciona directamente WordNet, resolviendo algunos problemas de referencias cruzadas que se plantean si utilizamos los synsets exclusivamente.

Mediante la representación de documentos por los hoods de los términos, se evita el problema de la representación por términos diferentes obtenidos a partir de sinónimos. Por lo que respecta a la polisemia, para asignar a una palabra un hood con un peso en un documento (o identificar su significado concreto en el contexto), Voorhees propone un criterio basado en la frecuencia de aparición de los hoods en la colección de documentos y en el texto considerado. De esta forma, dada una palabra en un documento y sus hoods asociados, le da mayor peso a los hoods que son más referenciados en el texto respecto al número de referencias en la colección. Este procedimiento de desambiguación, le permite asignar un mayor peso a los hoods que son más referenciados en un documento, y que deben representar mejor el significado concreto del término en el documento. Una vez representados los documentos por sus hoods con sus pesos, se define una función de similitud análoga a la del modelo del espacio vectorial.

Voorhees presenta una comparación de los resultados obtenidos por el sistema frente a uno basado en el modelo del espacio vectorial, pesos de términos y uso de extracción de raíces. Se utilizan para ello cinco colecciones de documentos y consultas estándar. A diferencia de los sistemas anteriores, los resultados son ventajosos para la aproximación más sencilla que para la basada en la utilización de la información léxica de WordNet presentada por Voorhees.

Desde nuestro punto de vista, el comportamiento de este sistema frente a UCF es análogo al de la utilización de thesaurus de propósito general y específicos del dominio, respectivamente. La información léxica existente en WordNet utilizada por Voorhees es demasiado general e introduce ruido en el proceso de recuperación frente a la aproximación seguida en UCF, en la que se utiliza información lingüística específica del dominio. Por otra parte, la obtención de esta información implica un menor esfuerzo de desarrollo en la aproximación de Voorhees.

Las causas que según el autor del trabajo explican los malos resultados de la evaluación son:

> *"Much of the degradation is due to the difficulty of disambiguating word senses in the short query statements: with little context to use in disambiguating, the indexing procedure selects an incorrect sense... (and) the query no longer matches documents in which the sense is correctly resolved... Nevertheless some queries do exhibit the performance improvements that conceptual retrieval suggest is possible."*

En resumen, el proceso de desambiguación por métodos estadísticos y léxicos se ve dificultado para las consultas en mayor medida que para los documentos debido a la brevedad de las primeras. Trabajos posteriores [Hearst94] indican que otras operaciones sobre documentos orientadas al acceso a la información pueden verse

mejoradas mediante la utilización de información léxica semántica independiente del dominio como la de WordNet. Estas operaciones se centran en el procesamiento, exclusivamente, de documentos de longitud suficiente para la utilización adicional de criterios estadísticos.

4.8 Resumen y conclusiones

Este capítulo comienza con una revisión del estado del arte en Procesamiento del Lenguaje Natural, centrada en los aspectos más relevantes para su integración en la Recuperación de Información. Esta revisión nos permite concluir que ha sido el avance experimentado en este campo durante los últimos años el que ha hecho posible la aparición de sistemas capaces de abordar problemas de PLN de dimensión real, tales como los sistemas de extracción de información. Dos aspectos centrales de este tipo de sistemas son su especificidad de dominio y su importante esfuerzo de desarrollo.

En la actualidad, los sistemas de PLN se apoyan en dos líneas maestras: la ampliación de los formalismos gramaticales mediante las gramáticas lógicas o las basadas en restricciones, y la importancia creciente del léxico hasta convertirse en la estructura que sirve de soporte a la mayor parte de la información semántica y lingüística de los sistemas.

En la primera línea, las gramáticas de unificación juegan, en nuestra opinión, el papel más relevante. Estos formalismos gramaticales constituyen la base de entornos altamente declarativos, en los que una gran cantidad de detalles algorítmicos no son necesarios de detallar para el desarrollo de sistemas de PLN. Se han estudiado DCG y PATR como los formalismos de este tipo más representativos, para los que en la actualidad existen herramientas que permiten el desarrollo de sistemas eficientes, especialmente el primero.

Con respecto a la segunda línea, se ha estudiado el papel que pueden jugar las grandes bases de conocimiento en los sistemas de PLN. Se han distinguido dos tipos: las bases de conocimiento de propósito general y las bases de datos léxicas. Las bases de conocimiento de propósito general, como Cyc, incluyen conocimiento de un amplio espectro de aspectos y en la actualidad presentan importantes problemas para su utilización práctica en sistemas de PLN. Los proyectos orientados al desarrollo de bases de datos léxicas se encuentran en un estado de desarrollo más avanzado y han permitido recientemente la consecución de sistemas completos como WordNet, con información de 95.600 términos, 70.100 conceptos y sus diferentes relaciones semánticas. Las bases de datos léxicas pueden ser la clave para el desarrollo de nuevos sistemas de PLN a escala real.

En nuestro trabajo tienen una especial importancia los sistemas de RI basados en el PLN. Para este tipo de sistemas, hemos distinguido dos aproximaciones: la basada en la sintáxis y la basada en la semántica. Las diferencias más importantes entre ambas son la dependencia del dominio, el coste de desarrollo y las mejoras en la efectividad proporcionadas en el proceso de recuperación.

Los sistemas basados en la sintáxis se centran en la obtención de frases de términos a partir del análisis sintáctico de los documentos (y las consultas), utilizándose diferentes relaciones sintácticas en cada sistema. La profundidad del análisis varía según los casos. En nuestro estudio hemos visto que los sistemas que siguen esta aproximación consiguen mejoras limitadas en la efectividad del proceso de recuperación.

Los sistemas basados en la semántica consiguen mejoras más importantes en la efectividad del proceso de recuperación. Como contrapartida, a diferencia de los sistemas basados en la sintaxis, utilizan información lingüística específica del dominio de los documentos que procesan y es necesario redefinirla si se desean trasladar a un dominio diferente, conllevando un mayor esfuerzo de desarrollo.

Dentro de la aproximación semántica hemos diferenciado dos tipos de sistemas, los basados en la interpretación del texto y los basados en la información léxica. Los primeros, como FERRET y NLDB, realizan un procesamiento de la documentación análogo al de sistemas como los de comprensión de texto o de extracción de información. Los últimos utilizan exclusivamente como información lingüística la correspondiente al nivel de léxico.

En sistemas que siguen la última aproximación, resulta posible la obtención de mejoras significativas en la efectividad incluyendo información léxica específica del dominio, como en UCF. En otro sistema, propuesto por Voorhees, se hace patente cómo la utilización de una base de datos léxica -WordNet- para la obtención de esta información puede disminuir el esfuerzo de desarrollo, pero presenta dificultades en su aplicación en la mejora del proceso de recuperación. En este sistema se utiliza la información semántica léxica de propósito general existente en WordNet, pero no se consiguen mejoras significativas en la efectividad. El tamaño de los ítems a analizar (documentos y consultas) se presenta como una cuestión determinante en la aproximación de Voorhees: su brevedad dificulta el buen funcionamiento del algoritmo de desambiguación en que se basa el sistema.

Todo este estudio nos ha permitido fijar la estrategia a seguir en la implementación del sistema Ares, que integra técnicas de PLN en el proceso de RI en un dominio concreto: el de descripciones de órdenes de UNIX. En Ares seguimos la aproximación de los sistemas basados en la semántica, explotando la información léxica obtenida de WordNet. En el sistema se incluye información lingüística específica del dominio mediante una gramática de unificación. El sistema se centra en los problemas que plantean las descripciones cortas de componentes software para las técnicas basadas en criterios estadísticos, consiguiendo mejorar la efectividad del proceso de recuperación. En el siguiente capítulo se presenta con detalle este sistema.

CAPITULO 5

EL SISTEMA ARES

5.1 Introducción

Ares es un sistema de Recuperación de Información orientado a la documentación existente en bibliotecas de componentes software. En concreto, Ares trabaja sobre la documentación asociada a las órdenes del sistema operativo UNIX, y representa una continuación del trabajo desarrollado en Argos. Para construir el sistema Ares se ha desarrollado un modelo que integra técnicas de Procesamiento del Lenguaje Natural en el proceso de recuperación de documentos. De esta forma se ha conseguido mejorar significativamente la efectividad del proceso de recuperación.

En el desarrollo del sistema Ares se ha seguido una aproximación basada en la semántica. La aproximación basada en la semántica permite mejorar la efectividad del proceso de recuperación, pero los sistemas desarrollados siguiendo esta aproximación son, intrínsecamente, específicos de un dominio concreto y presentan como principal inconveniente un alto coste de desarrollo. En el desarrollo de Ares hemos partido de un dominio concreto buscando mejorar la efectividad del proceso de recuperación, pero intentando minimizar el esfuerzo de desarrollo.

El dominio para el que se ha desarrollado Ares es, como en el caso del sistema Argos, la información contenida en el manual del sistema operativo UNIX. El sistema Ares utiliza conocimiento específico del sistema operativo UNIX, así como del lenguaje utilizado en el manual. Si bien Ares se ha desarrollado específicamente para este dominio, la aproximación seguida en él puede ser, como veremos, válida para el tratamiento de la documentación existente en otras bibliotecas de componentes software. De hecho, Ares intenta resolver un problema concreto que presenta el tipo de documentación existente en ciertas bibliotecas de componentes software, y que hemos denominado "el problema de las descripciones cortas".

En el desarrollo de Ares hemos tenido presentes dos objetivos fundamentales. El primero ha sido conseguir reducir, mediante el uso de técnicas de PLN, los problemas que presenta una aproximación basada en el modelo vectorial, extracción de raíces y utilización de listas de palabras vacías (y en general los métodos basados en criterios estadísticos) para el caso de las "descripciones cortas". El segundo

objetivo es conseguir un sistema cuyo coste de desarrollo sea significativamente inferior al de los sistemas de recuperación de información basados en el procesamiento del lenguaje natural que siguen la aproximación semántica. Estos objetivos se han podido lograr gracias a la reutilización de la información de la base de datos léxica WordNet, y a la utilización de una gramática de unificación como base para la implementación del analizador-traductor.

En este capítulo comenzamos analizando el problema de las descripciones cortas para después pasar a describir el planteamiento y la estructura del sistema Ares. En puntos sucesivos se analizan y describen con detalle cada uno de los elementos que componen el sistema:

- El léxico del sistema, obtenido automáticamente de WordNet.
- El lenguaje de representación de las descripciones.
- La gramática de unificación.
- El cálculo de la similitud.

Finalmente presentamos una serie de experimentos centrados en la efectividad y orientados a evaluar el comportamiento del sistema frente a las aproximaciones basadas en el modelo vectorial, uso de extracción de raíces, lista de parada y thesaurus.

5.2 Las descripciones cortas de componentes software

Existe un número importante de bibliotecas de componentes software que incluyen documentación en lenguaje natural, con descripciones de las componentes mucho más breves que los documentos inicialmente considerados en el sistema Argos (las descripciones de las órdenes correspondientes a la sección 1 del manual del sistema operativo UNIX). Desde el punto de vista de la recuperación de información, la longitud reducida de los documentos plantea una serie de problemas. Trataremos a continuación esta cuestión y el subconjunto de la documentación del sistema operativo UNIX que hemos considerado en el desarrollo de Ares, como representante del caso de las descripciones cortas.

5.2.1 El problema de las descripciones cortas

El tamaño de las descripciones en Lenguaje de Natural que proporcionan las colecciones de componentes software puede variar significativamente de unos casos a otros. Los documentos o descripciones en lenguaje natural de las 525 órdenes existentes en la sección 1 del manual del sistema operativo UNIX (SunOS 4.1.3) [Sun89] tienen una extensión considerable: su longitud media es del orden de las 1000 palabras. Sin embargo, existe un número importante de casos en los que las colecciones disponen de descripciones en Lenguaje Natural asociadas a cada componente de una longitud mucho menor. Un caso concreto de este tipo de colecciones lo constituye, dentro del mismo sistema operativo UNIX, la biblioteca de funciones en C que es descrita en la sección 3 del manual [Sun89]: el sistema operativo proporciona al usuario más de 1000 funciones básicas, y la descripción de

add: element
"Add a new <element> to the receiver, returning the added element."
addAll: aCollection
"Add each element of <aCollection> to the receiver, returning the collection of elements that were added."
asArray
"Answer an Array containing all the elements of the receiver."
isEmpty
"Answer <true> if the receiver contains no elements."
reject: conditionBlock
"Answer a collection whose elements are the elements of the receiver for which the <conditionBlock> evaluates to <false>. The condition block will be evaluated once for each element of the receiver."

char *strncat(s1, s2, n)
char *s1, *s2;
int n;
strcat() appends a copy of string s2 to the end of string s1. strncat() appends at most n characters. Each returns a pointer to the null-terminated result.
int strlen(s)
char *s;
strlen() returns the number of characters in s, not including the null-terminating character.
char *strpbrk(s1, s2)
char *s1, *s2;
strpbrk() returns a pointer to the first occurrence in string s1 of any character from string s2, or a NULL pointer if no character from s2 exists in s1.
char *strstr(s1, s2)
char *s1, *s2;
strstr() returns a pointer to the first occurrence of the pattern string s2in s1. If s2 does not occur in s1, strstr() returns NULL.

Figura 5.1: Descripciones cortas de componentes software

cada una de ellas tiene una longitud media aproximada de 30 palabras. Otro ejemplo es el de la jerarquía de clases incluida en el entorno Smalltalk V.5 [Digitalk93], formada por más de 100 clases y más de 1000 métodos con descripciones de cada método de una longitud media del orden de 20 palabras. En la figura 5.1. se presenta una muestra de algunas descripciones de este tipo, correspondientes a métodos de la clase Collection, en el entorno Smalltalk V.5, y a funciones en C para el manejo de cadenas de caracteres en UNIX.

La longitud de los documentos puede ser una cuestión que determine la efectividad de una aproximación a la recuperación de información [Lewis92, Voorhees93, Masand93, Salton93, Hearst94]. Las aproximaciones basadas en criterios estadísticos presentan limitaciones intrínsecas en el caso de las descripciones cortas. Casos concretos los constituyen la dificultad para la obtención automática de thesaurus [Salton83, Strinivasdan92], o los algoritmos de clustering de documentos [Salton83, Rasmussen92]. En general, la efectividad de las operaciones basadas en criterios estadísticos se ve disminuida de forma importante cuando en los cálculos que se realizan, el número de palabras, o términos, de los documentos que intervienen es pequeño, como en el caso que estamos discutiendo.

De forma cuantitativa, una cuestión importante es la disminución del *recall*. Es decir, no se logra recuperar una proporción importante de documentos potencialmente relevantes a la necesidad del usuario [Salton83]. En el caso de la aproximación basada en el modelo vectorial, extracción de raíces y uso de listas de parada, se puede presentar una razón como inmediata: cuanto menor es la longitud

DF(1V) USER COMMANDS DF(1V)

NAME

df - report free disk space on file systems

SYNOPSIS

df [-a] [-i] [-t type] [filesystem...]
[filename...]

SYSTEM V SYNOPSIS

/usr/5bin/df [-t] [filesystem...] [filename...]

AVAILABILITY

The System V version of this command is available with the System V software installation option. Refer to Installing SunOS 4.1 for information on how to install optional software.

DESCRIPTION

df displays the amount of disk space occupied by currently mounted file systems, the amount of used and available space, and how much of the file system's total capacity has been used. Used without arguments, df reports...

Figura 5.2: Fragmento de documentación de orden del manual de UNIX.

de un texto, menor es el número de términos que lo representan y menor la probabilidad de que aparezcan entre ellos los términos utilizados en la consulta del usuario. Relacionado con esta cuestión, un problema que cobra especial relevancia es el de la sinonimia. El usuario puede utilizar términos en la consulta que son sinónimos de los que se incluyen en el documento. Estos términos sinónimos, diferentes, pero con igual significado, no son considerados en el cálculo de la similitud. De acuerdo con esto, en una primera aproximación, la utilización de información de diccionarios de sinónimos se presenta especialmente adecuada en este caso.

5.2.2 Las descripciones cortas de las órdenes de UNIX

En el desarrollo de Ares hemos fijado como dominio las descripciones cortas de las órdenes de UNIX. En concreto, las existentes en la sección 1 del manual del sistema operativo UNIX (SunOS 4.1.3). La descripción corta de una orden se encuentra bajo el epígrafe *name*, en el comienzo de la página del manual correspondiente. En la figura 5.2. aparece el comienzo de la página del manual correspondiente a la orden *df*, que proporciona el espacio libre que existe el sistema de archivos ("report free disk space on file systems"). La descripción corta de una orden puede estar formada por una o dos frases en las que se describe su funcionalidad de forma breve y concisa.

Estas descripciones se pueden encontrar también agrupadas en el archivo de introducción del manual (*man intro*). Este conjunto de descripciones cortas está formado por 479 elementos. En la figura 5.3 presentamos algunas de las descripciones cortas disponibles en el manual. De las 479 descripciones hemos eliminado las 47 correspondientes a las órdenes internas del C-Shell que incluyen

at	at(1)	execute a command or script at a specified time
cflow	cflow(1V)	generate a flow graph for a C program
chgrp	chgrp(1)	change the group ownership of a file
chmod	chmod(1V)	change the permissions mode of a file
clear	clear(1)	clear the terminal screen
cut	cut(1V)	remove selected fields from each line of a file
df	df(1V)	report free disk space on file systems
diff	diff(1)	display line-by-line differences between pairs of text files
egrep	grep(1V)	search a file for a string or regular expression
hostname	hostname(1)	set or print name of current host system
iconedit	iconedit(1)	create and edit images for icons, cursors and panel items
lastcomm	lastcomm(1)	show the last commands executed, in reverse order
look	look(1)	find words in the system dictionary or lines in a sorted list
mv	mv(1)	move or rename files
on	on(1C)	execute command on a remote system with local environment
rm	rm(1)	remove (unlink) files or directories

Figura 5.3: Descripciones cortas de órdenes de UNIX.

como descripción una referencia cruzada que no describe la orden en sí misma (para todas ellas la descripción es "C shell built-in commands, see csh.1").

El conjunto de descripciones considerado consta en total de 432 elementos, la longitud media de la descripción es de 6,39 palabras, y el léxico utilizado en ellas está formado por 661 palabras diferentes. En el apéndice final se presentan la colección de descripciones utilizada, representada en formato de hechos Prolog. La colección representada de esta forma se ha obtenido mediante la implementación de un preprocesador del manual, que describiremos en puntos siguientes.

Aunque Ares es un sistema construido específicamente para esta colección de descripciones, los criterios seguidos en su desarrollo resultan de inmediata adaptación a otras colecciones de componentes. En concreto, también ha sido desarrollado un analizador basado en Ares para las descripciones cortas de las "recetas" del "VisualWorks Cookbook" [González95]. En estas 515 descripciones se tratan las formas de implementación de las tareas más frecuentes sobre el entorno VisualWorks, basado en SmallTalk [Parc94].

La selección de las descripciones cortas de las órdenes de UNIX como dominio concreto en Ares, se ha debido a diferentes razones. En primer lugar, proporciona una mayor continuidad respecto al trabajo desarrollado en Argos y nos permite reutilizar gran parte de la experiencia adquirida sobre el dominio. Además, es una colección de componentes lo suficientemente representativa del caso de las descripciones cortas, tanto por el número de descripciones como por su longitud. Otra razón importante es que su elección ha facilitado el desarrollo de una serie de experimentos centrados en la efectividad. Estos experimentos han servido para la evaluación e interpretación del comportamiento del sistema y de nuestra aproximación.

5.3 Objetivos y estructura del sistema

En Ares nos hemos marcado dos objetivos esenciales:

1. Construir un sistema basado en PLN que proporcione mejoras en la efectividad del proceso de recuperación para el caso de las descripciones cortas. Entendiendo que la mejora es introducida respecto a una aproximación basada en el modelo vectorial, uso de pesos de términos, lista de parada y extracción de raíces.

2. Desarrollar un sistema de recuperación de información basado en PLN que conlleve un bajo coste o esfuerzo de desarrollo.

Por lo que respecta a la mejora en la efectividad, desde un punto de vista general, una aproximación basada en el PLN puede resolver las limitaciones intrínsecas que presentan los métodos basados en criterios estadísticos cuando se aplican a las descripciones cortas. La información que los métodos estadísticos obtienen a partir de los textos -como el caso ya mencionado de la generación automática de thesaurus- puede ser sustituida por la información lingüística que se incluye en un sistema basado en PLN durante la fase de desarrollo. Esta información lingüística se obtiene mediante un análisis manual del dominio en la mayoría de los sistemas de RI basados en PLN que siguen la aproximación semántica, lo que implica un importante esfuerzo de desarrollo.

Esta última cuestión nos conduce al segundo objetivo de nuestra aproximación: reducir el esfuerzo de desarrollo del sistema. Un problema importante y típico de los de los sistemas PLN. Los dos puntos principales en los que nos hemos basado para resolver este problema son:

1. Utilización de información de una base de conocimiento léxico ya existente, en concreto, WordNet.

2. Utilizar información lingüística específica del dominio, que es incluida en el sistema en una gramática de unificación que facilita su desarrollo.

En nuestra aproximación, el proceso de recuperación se concibe de una forma esencialmente análoga a la forma general presentada en capítulos anteriores (punto 2.2, figura 2.1). La novedad consiste en la adición de un elemento determinante en el proceso: el conocimiento lingüístico (figura 5.4). El conocimiento lingüístico que utiliza el sistema se divide a su vez en dos grandes núcleos: el conocimiento gramatical y el conocimiento léxico. Disponer de este conocimiento adicional es lo que, desde una visión de alto nivel, permite a Ares lograr un incremento en su efectividad.

Conviene también señalar los aspectos en los que se puede establecer un cierto paralelismo entre la aproximación seguida en Ares, basada en el PLN, y la utilizada en Argos, basada en el modelo del espacio vectorial, uso de pesos de términos, extracción de raíces y lista de parada. En primer lugar, desde un punto de vista general, la aproximación seguida en Argos podría también adaptarse al esquema

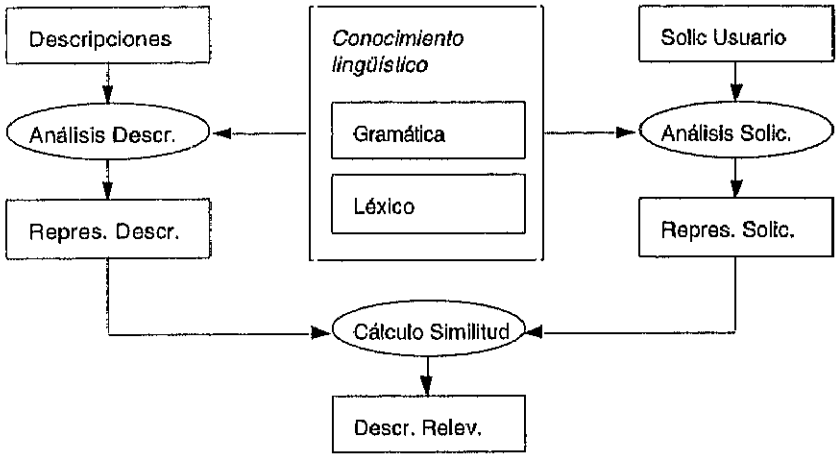


Figura 5.4: El proceso de recuperación en Ares

que representa el proceso de recuperación de la figura 5.4, pero teniendo presente la limitación del conocimiento lingüístico de que hace uso: el método de extracción de raíces -conocimiento esencialmente morfológico- y la lista de parada -conocimiento léxico muy reducido-. Por otra parte, en la aproximación seguida en Ares, el conocimiento lingüístico juega un papel fundamental, pero no se renuncia a la utilización de criterios estadísticos. De hecho, incluimos elementos basados en frecuencias de aparición de términos en el cálculo de la similitud, como se tratará en puntos siguientes.

Los aspectos fundamentales de nuestra aproximación y que han constituido la base para el desarrollo de Ares, son:

- La construcción del léxico del sistema a partir de WordNet.
- La definición de un lenguaje de representación de las componentes y las consultas
- La gramática de unificación
- El algoritmo del cálculo de la similitud

La información léxica utilizada por Ares se obtiene de WordNet de forma automática. La información gramatical utilizada por Ares se encuentra almacenada en la gramática de unificación de que hace uso el sistema y se ha desarrollado para este dominio específico. La gramática y el léxico son utilizados tanto para realizar el análisis de descripciones como para analizar las consultas de los usuarios.

La estrategia general seguida en nuestro sistema se centra en utilizar la información lingüística a nivel de léxico existente en WordNet, que no es específica de dominio, esencialmente como mecanismo orientado a la mejora del *recall*. La información lingüística específica del dominio, incluida en la gramática de unificación, se encuentra orientada fundamentalmente a la mejora de la *precision*.

```
command(
     name('apropos'),
     description(['locate','commands','by','keyword','lookup']) ).
command(
     name('ar'),
     description(['create','library','archives',',','and','add',
          'or','extract','files']) ).
command(
     name('arch'),
     description(['display','the','architecture','of','the',
          'current','host']) ).
command(
     name('at'),
     description(['execute','a','command','or','script','at','a',
          'specified','time']) ).
command(
     name('atq'),
     description(['display','the','queue','of','jobs','to','be',
          'run','at','specified','times']) ).
command(
     name('atrm'),
     description(['remove','jobs','spooled','by','at','or',
          'batch']) ).
```

Figura 5.5: Fragmento de base de datos Prolog con descripciones de órdenes.

La representación de las solicitudes y de los documentos es diferente a la utilizada en Argos. En Ares, la forma de representación de documentos y solicitudes es de tipo *frame*, y es análoga a las utilizadas en los sistemas basados en la semántica tales como FERRET. En el cálculo de la similitud se comparan la representación de la consulta del usuario con las representaciones de los documentos.

Respecto a la implementación del sistema, Ares es un sistema uniforme, con todos sus módulos codificados en Prolog, así como todos los datos con los que trabaja. Es decir, tanto el conjunto de documentos y solicitudes de usuarios, como la información léxica y gramatical utilizada, están representados como conjuntos de predicados y hechos Prolog. Se han desarrollado diversos preprocesadores implementados en C y lex [Ritchie88, Sun88] que han traducido la información de partida (fundamentalmente, la existente en el manual del sistema UNIX y la existente en WordNet) a la sintaxis de Prolog. En la figura 5.5 se presenta una muestra de predicados Prolog correspondientes a descripciones de componentes que se han obtenido a partir de los archivos originales del manual mediante un preprocesador desarrollado en lex. Los distintos módulos en Prolog del sistema se han desarrollado simultáneamente en SICStus Prolog [Sicstus93] en una estación Sun Sparc 10 y en AAIS Prolog [AAIS91] en un Apple Macintosh PowerPC, explotando las ventajas de cada entorno

Esta aproximación nos ha permitido disminuir el esfuerzo del desarrollo del sistema facilitando la integración de los diferentes módulos y, además, nos ha facilitado la adquisición de conocimiento lingüístico específico de nuestro dominio. Adquirir este conocimiento del lenguaje utilizado en las descripciones ha facilitado la definición

de elementos del sistema, principalmente de la gramática. Disponer de las descripciones, accesibles directamente mediante consultas al intérprete de Prolog, facilita la adquisición de conocimiento [Clayton92] de datos concretos como, por ejemplo, el número de descripciones que incluyen un determinado término, o saber cuáles incluyen más de un verbo o en cuáles no aparece ninguno. En general, facilita el desarrollo incremental de la gramática. Poder saber de forma inmediata el número de descripciones analizadas correctamente nos permite conocer la cobertura del analizador. Identificar las descripciones que no son analizadas correctamente, permite utilizarlas como base para ampliaciones de la gramática. Esta cuestión será discutida con más detalle en puntos siguientes.

Los elementos principales que constituyen la arquitectura del sistema se corresponden con los aspectos fundamentales de nuestra aproximación. En la figura 5.6 se incluye el Diagrama de Flujo de Datos (DFD) [Yourdon79, Rumbaugh91, Pressman93] que describe la arquitectura de Ares. En el diagrama modificamos la notación habitual para dar cuenta de las peculiaridades del sistema: representamos las unidades de procesamiento mediante círculos (o "burbujas") en la forma habitual, pero representamos los elementos de datos que constituyen las entradas/salidas de las unidades de procesamiento mediante cuadrados. Los elementos sombreados se han desarrollado manualmente, mientras que los no sombreados se encontraban ya disponibles o han sido obtenidos de forma automática. Los sufijos que aparecen en los elementos (*.pro, .c, .lex*) indican la forma en que se encuentran almacenados o en que han sido codificados los datos.

Las fuentes de información de que se parte son dos: los archivos de texto del manual y los correspondientes a la base de datos de WordNet. También se parte de un archivo codificado en Prolog con las consultas del usuario que van a ser procesadas. Dada la condición experimental del sistema, no suponemos una interacción directa con el usuario, sino que partimos de la existencia de una base de datos con el conjunto de consultas. Esta cuestión se detallará en el punto de este capítulo dedicado a la exposición de los experimentos centrados en la efectividad que hemos realizado con el sistema.

Para la traducción de las descripciones del manual a hechos Prolog se ha desarrollado un preprocesador en lex [Sun88], *preprocman.lex*. El módulo del sistema, *generalexico.pro.lex.c*, tiene como entrada los archivos en que se almacenan originalmente los datos en WordNet, las descripciones representadas en Prolog (*man.pro*) y el archivo de consultas del usuario (*consultas.pro*). Este módulo selecciona el subconjunto de la información léxica existente en WordNet que utilizaremos para el análisis semántico de las descripciones del manual y las consultas, y genera el archivo que contiene el léxico del sistema (*lexico.pro*).

El análisis del manual (*analizardescrip.pro*) y las consultas de los usuarios (*analizarconsultas.pro*) utiliza la gramática de unificación (*gramatica.pro*) y el léxico obtenido de WordNet. Los resultados de ambos análisis se almacenan en dos archivos Prolog: *analisisdescrip.pro* y *analisisconsultas.pro*, respectivamente.

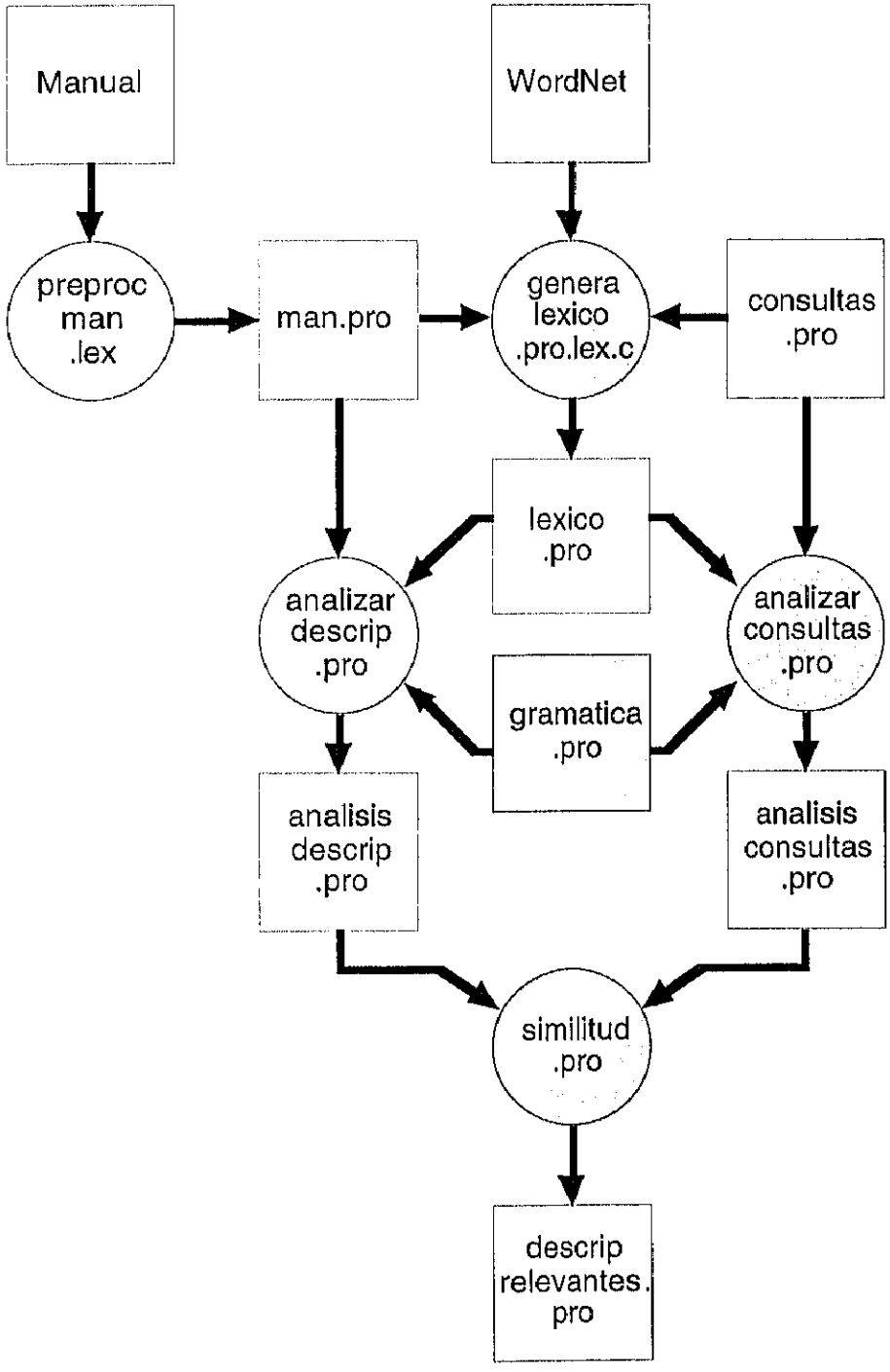


Figura 5.6: Diagrama de Flujo de Datos de Ares.

Los análisis de documentos y consultas constituyen la entrada del proceso de cálculo de la similitud (*similitud.pro*), que genera un archivo con los valores numéricos calculados de la similitud o relevancia de las descripciones a cada consulta (*descriprelevantes.pro*).

En los puntos siguientes tratamos con detalle los problemas específicos de la implementación del sistema. Detalles adicionales de codificación se incluyen en el apéndice final.

5.4 Construcción del léxico del sistema

En WordNet se encuentra almacenada una gran cantidad de información léxica: 95.600 términos organizados en 70.100 synsets o conceptos para los que se incluye información referente a diferentes relaciones semánticas. Estas relaciones son principalmente las de sinonimia, antonimia, meronimia e hiponimia, constituyendo en su conjunto una base de datos de 12 MegaBytes en la versión 1.4 [Miller93]. Toda esta información léxica -lingüística- puede ser utilizada para mejorar el proceso de recuperación.

Para construir el léxico utilizado por Ares, hemos seleccionado un subconjunto de la información existente en WordNet y la hemos traducido a la sintaxis de Prolog. Esta aproximación ha sido necesaria, no como mero aspecto de eficiencia del sistema, sino como cuestión determinante de la factibilidad de Ares. WordNet es una base de datos de grandes dimensiones que está diseñada para su consulta de forma interactiva por usuarios y los tiempos de acceso a la información son considerables (en nuestro sistema son necesarios un gran número de accesos al léxico durante el proceso de análisis). Por otra parte, los interfaces que se proporcionan para el acceso a la base de datos no son lo suficientemente flexibles como para disponer, efectivamente, de toda la información existente en ella. Disponer del subconjunto de la información de WordNet útil para el procesamiento que realiza Ares en su léxico, en forma de base de datos Prolog, facilita en gran medida el proceso de desarrollo del sistema y posibilita su utilización desde la gramática de unificación que constituye la base del analizador.

Para la selección del subconjunto de la información de WordNet que incorporamos en el léxico de Ares, hemos tenido presentes dos criterios generales. En primer lugar, del conjunto de términos para los que WordNet incluye información (95.600) sólo es necesaria la información existente en la base de datos relativa al lenguaje que Ares debe procesar: los términos utilizados en las descripciones del manual y en las consultas de los usuarios. En segundo lugar, entre la información referente a todas las relaciones léxicas incluidas en WordNet, se ha seleccionado aquella que resulta especialmente adecuada para el procesamiento de descripciones y consultas en nuestra aproximación.

En el léxico de Ares hemos incluido dos tipos de información tomada de WordNet: información sintáctica e información semántica. Para cada palabra que aparece en las descripciones, hemos obtenido su categoría sintáctica y el conjunto de synsets, o conceptos, en los que se encuentra incluida en WordNet. En la figura 5.6 se

```
word(['about'],adje,[00155141]).
word(['about'],adve,[00019458,00019360,00004330]).
word(['access'],noun,[02750694,01637511]).
word(['access'],verb,[00897204,00809187]).
word(['accord'],noun,[06232515,03509380,03343660,02582438]).
word(['accord'],verb,[01066472,00900348]).
word(['account'],noun,[05962020,05961102,05941089,03526317,03331309,03309943,0324079,
      03238974]).
word(['account'],verb,[00904682]).
word(['accounting'],noun,[05962020,05961102,02974696,00219964]).
word(['activity'],noun,[06242704,02552560,00140517]).
word(['add'],verb,[00928104,00541380,00411901,00376836,00246326,00212604,00073989]).
word(['adjacent'],adje,[00273234,00217177]).
word(['administer'],verb,[00962615,00922236,00035306]).
word(['administration'],noun,[03962855,00386521,00237828]).
word(['advance'],adje,[00396097,00115410]).
word(['advance'],noun,[06131942,05946493,03580790,03574387,03504843,00101359]).
word(['advance'],verb,[00948498,00911827,00803595,00803475,00802989,00440447,0020991,
      00101998]).
word(['advanced'],adje,[01116968,00937639,00918761,00763539,00608320,00397918]).
word(['affirmative'],adje,[01337502,00044990,00492970]).
word(['affirmative'],noun,[03519477]).
word(['alarm'],noun,[03639308,03355896,01649059,01648368]).
word(['alarm'],verb,[00712224,00343899]).
word(['alias'],noun,[03184083]).
word(['all'],adje,[01116032]).
word(['allow'],verb,[00958585,00900470,00319663,00312395,00281882,00281463]).
word(['alter'],verb,[00671535,00081649,00050777,00049698,00027518,00027158]).
word(['analysis'],noun,[03195965,03100627,03090759,03012330,00238291,00228911]).
word(['another'],adje,[01034661,00777484]).
word(['application'],noun,[03238149,02624882,00320406,00242015,00225461]).
word(['arbitrary'],adje,[00350021,00349328]).
word(['architecture'],noun,[03120909,01663179,00217919]).
.../...
word(['do'],noun,[03605098]).
word(['do'],verb,[01071505,01055769,01034508,01018225,01015998,01015621,00738915,
      00691099,00664342,00654930,00018689]).
.../...
word(['execute'],verb,[01016285,01015621,00984018,00983489,00691099,00662251,
      00397674]).
.../...
word(['perform'],verb,[00939855,00691616,00691099]).
word(['perform'],verb,[00939855,00691616,00691099]).
word(['performance'],noun,[03579376,03400945,03005711,00192709,00039771]).
word(['perhaps'],adve,[00051807]).
word(['permission'],noun,[06139050,03313658,00388084]).
word(['permit'],noun,[06139050,03313984,01496572]).
word(['permit'],verb,[00312395]).
word(['permute'],verb,[00161500]).
word(['physical'],adje,[01301241,00889154,01056599,00886289,00790731,00304907]).
word(['piece'],noun,[05906768,04296727,03715044,03560920,03459205,03159908,02369832,
      02258747,02044212,01828046,01573798]).

...
```

Figura 5.6. Muestra del léxico de Ares

presenta una muestra del léxico del sistema. La confección del léxico se ha conseguido mediante la construcción de un programa Prolog para la generación del corpus de palabras de las descripciones y las consultas, otro en C que gestiona la consulta a WordNet, y varios filtros en lex que traducen los datos a la sintaxis de Prolog. Los detalles de esta implementación se proporcionan en el apéndice final. La base de datos Prolog que constituye el léxico utilizado por Ares para el procesamiento de las descripciones está formada por:

- 845 hechos
- 641 términos
- 2843 conceptos o synsets

Palabras diferentes como "copy", "copies", "copying", o "copied" son representadas por un sólo término, "copy", que tiene diferentes categorías sintácticas, nombre y verbo, cada una de las cuales tiene diferentes significados.

A continuación describimos con más detalle los dos tipos de información incluida en el léxico del sistema: la información semántica y la información sintáctica.

5.4.1 Información semántica

En el léxico de Ares hemos incluido los posibles significados de cada término que se encuentran definidos en WordNet. En WordNet cada significado, o synset, se identifica unívocamente por un entero de ocho dígitos. Cada significado en WordNet se encuentra también descrito -pero no identificado unívocamente mediante una breve descripción en lenguaje natural. Ya que cada significado puede encontrarse asociado con varias palabras, un significado también puede ser descrito mediante éstas.

Como ejemplo, la palabra "execute" tiene siete significados, los siete como verbo o acción (figura 5.6). Los siete significados son identificados por los enteros [01016285, 01015621, 00984018, 00983489, 00691099, 00662251, 00397674]. Las descripciones de estos siete significados son:

```
01016285:    execute -- (carry out the legalities of)
01015621:    execute, make, do, effect, carry_out -- (carry or
             put into effect; e.g. "make an effort"; "do research")
00984018:    execute -- (murder execution-style)
00983489:    execute, put_to_death -- (socially sanctioned killing
             as a means of punishment)
00691099:    perform, execute, do -- (recreate, create again)
00662251:    complete, accomplish, execute, carry_out, fulfill --
             (cause   to happen; carry through)
00397674:    execute -- (sign in the presence of witnesses)
```

Disponer de un conjunto de significados como este facilita la representación semántica de las expresiones en lenguaje natural y permite resolver los problemas que se plantean en las aproximaciones a la recuperación de información que utilizan una indexación -o representación de documentos y consultas- basada en términos. Trataremos a continuación cómo se afrontan en Ares, haciendo uso de esta información semántica, los problemas derivados de la polisemia y la sinonimia.

Para aproximaciones que utilizan una indexación basada en términos obtenidos a partir de palabras, la polisemia conlleva problemas. En concreto, para expresiones en las que aparece un mismo término, pero con significados diferentes, este contribuye a un incremento de la similitud -cosa no deseable-. Por ejemplo, si consideramos las dos siguientes expresiones:

"execute a command" (execute -> 00691099)
"execute the criminal" (execute -> 00983489)

En una aproximación que utilice una indexación basada en palabras, la aparición de "execute" en ambas expresiones, conllevaría un incremento en el valor de su similitud. En una representación basada en los significados, la palabra es representada de forma diferente para cada expresión. En una aproximación a la recuperación de información, este tipo de consideraciones pueden utilizarse para conseguir fundamentalmente un mecanismo de mejora de la *precision* [Mauldin91, Voorhes93].

La sinonimia también presenta problemas en una aproximación que utiliza una indexación con términos basados en las palabras. En concreto, para expresiones en las que aparecen diferentes palabras, pero con un mismo significado, éstas no contribuyen con un incremento en el valor de la similitud. Por ejemplo consideremos las dos siguientes expresiones:

"execute an action" (execute -> 00691099)
"perform an action" (perform -> 00691099)

En ambas, "execute" y "perform" tienen un mismo significado (00691099). En una aproximación que utilice una indexación basada en términos, las palabras "execute" y "perform" no incrementan el valor de la similitud. Si en la representación se utilizan los significados de los términos, pueden ser considerados en el cálculo de la similitud y hacer que esta aumente. Este tipo de consideraciones pueden tenerse presentes para conseguir un mecanismo orientado a mejorar el *recall* [Mauldin91, Voorhes93].

De acuerdo con lo ya señalado en párrafos anteriores, en Ares utilizaremos la información semántica basada en la definición de conceptos (representados por sus identificadores numéricos) y su asociación a términos, como base para la mejora del *recall*.

El problema de la sinonimia cobra una especial relevancia en el caso de las descripciones. En general, cuanto menor es la longitud de un texto relevante, mayor es la probabilidad de que no aparezcan los términos de la consulta en él (aunque sí lo haga un sinónimo de alguno de estos términos) [Salton93]. Por otra parte, la solución clásica para este problema, el uso de thesaurus, se ve dificultada.

Podemos establecer un paralelismo entre las agrupaciones de términos incluidas en un thesaurus (o las listas de asociación de términos tratadas en capítulos anteriores) y la información semántica disponible en el léxico de Ares. En Ares, dos palabras tienen un mismo significado si ambas tienen asociado un mismo concepto. Como ejemplo tomado del léxico de Ares (figura5.6), "do", "execute" y "perform" son sinónimos porque comparten el synset o concepto con identificador 00691999:

```
[account,accounting,5961102]
[account,accounting,5962020]
[account,calculate,904682]
[account,history,3238974]
[account,report,3309943]
[accounting,account,5961102]
[accounting,account,5962020]
[allow,permit,312395]
[alter,change,49698]
[alter,change,50777]
[alter,interpolate,81649]
[alter,rename,49698]
[alter,rename,50777]
[another,different,1034661]
[archive,archives,1664058]
[archives,archive,1664058]
[arrive,get,808665]
[as,like,53614]
[ask,request,290892]
[base,common,799416]
[base,establish,244941]
[base,low,441780]
[base,low,799416]
[base,send,433311]
[basic,root,926198]
[basic,simple,925222]
[be,form,1035676]
[be,make,1035676]
[be,work,967974]
[become,form,1036766]
[become,get,62187]
[become,turn,1037421]
[bit,bite,3665703]
[bit,piece,3560920]
[bite,bit,3665703]
[block,stop,1014997]
[block,stop,1737603]
[build,construct,667242]
[build,form,2754975]
[build,frame,2754975]
[calculate,account,904682]
[call,name,386780]
[call,name,412619]
[cancel,delete,633190]
[change,alter,49698]
[change,alter,50777]
[change,convert,66589]
[change,exchange,66589]
[change,exchange,901469]
[change,exchange,6163449]
[change,modification,3555614]
[change,rename,49698]
[change,rename,50777]
[change,rename,66589]
[character,part,3064213]
[character,reference,3315504]
[characteristic,feature,3039127]
[chart,graph,3445254]
[chart,map,276032]
[check,contain,450786]
[check,contain,994472]
[check,control,255573]
[check,control,994472]
[check,mark,255471]
[check,screen,1003435]
[check,see,255573]
[check,service,255279]
[check,stop,450786]
[check,stop,6244919]
[check,tab,3241608]
[check,test,1003435]
[check,test,2997577]
[clear,free,814154]
[clear,make,914564]
[clear,make,915053]
[clear,smooth,135850]
[clear,strip,79069]
[clock,time,206957]
[code,encode,396075]
[code,encrypt,396075]
[code,protocol,3038222]
[combine,merge,167664]
[command,control,966130]
[command,control,2965262]
[command,order,3506089]
[common,base,799416]
[common,low,799416]
[common,simple,837632]
[compact,compress,567732]
[compact,compress,568261]
[compact,summary,264486]
[compare,see,108802]
[compress,compact,567732]
[compress,compact,568261]
[connect,join,1036370]
[connect,link,277336]
[connect,link,553185]
[connect,link,1036370]
[connect,relate,277336]
[construct,build,667242]
```

Figura 5.7: Asociaciones término-término-concepto obtenidas del léxico.

```
word(['do'],verb,[01071505,01055769,01034508,01018225,01015998,
        01015621,00738915,00691099,00664342,00654930,00018689]).
word(['execute'],verb,[01016285,01015621,00984018,00983489,
        00691099,00662251,00397674]).
word(['perform'],verb,[00939855,00691616,00691099]).
```

En la figura 5.7 incluimos las cien primeras asociaciones [*término*$_1$, *término*$_2$, *concepto*] de las 636 que se obtienen del léxico de Ares.

Este tipo de información, especialmente necesaria en el caso de las descripciones cortas, no puede ser obtenida mediante métodos de generación automática de thesaurus, cuando la información de partida son documentos con un número reducido de términos [Salton83]. Por lo tanto, en el caso de las descripciones cortas resulta necesario la confección manual de un thesaurus de términos. La construcción manual de un thesaurus es una labor costosa. Este mismo alto coste de desarrollo conlleva la confección de este tipo de información en los sistemas basados en el PLN que siguen la aproximación basada en la semántica, como FERRET o UCF. En el desarrollo de Ares hemos utilizado como información de este tipo la proporcionada por WordNet. El principal problema que presenta esta información obtenida de WordNet frente a un thesaurus específico del dominio o la de los sistemas basados en la semántica es su generalidad. La generalidad de esta información hace que su utilización conlleve una disminución de *precision* mayor que la introducida por un thesaurus específico del dominio, o la de sistemas basados en la semántica.

A diferencia del léxico, el resto de los elementos de Ares, que presentamos en puntos siguientes de este capítulo, sí son específicos del dominio. Estos elementos específicos del dominio introducir mecanismos orientados principalmente a la mejora de la *precision*. Es decir, conseguir rechazar aquellos documentos que se considerarían relevantes sin serlo realmente, utilizando exclusivamente la información relativa a la sinonimia. La conjunción de estos elementos con el léxico conlleva, por lo tanto, una mejora simultánea en *recall* y *precision*, y en la efectividad en general.

5.4.2 Información sintáctica

A partir de la información existente en WordNet, hemos incluido en el léxico, para cada palabra, sus posibles categorías sintácticas. Las categorías sintácticas consideradas son las distinguidas en WordNet: verbo, nombre, adjetivo y adverbio. Una palabra puede tener asociadas varias categorías sintácticas, como los casos de "access" o "archive" que pueden ser nombres o verbos. En este caso, incluimos un hecho Prolog en el léxico por cada posible categoría sintáctica. En general, una palabra tendrá diferentes synsets, o conceptos, asociados a cada categoría sintáctica, ya que sus significados son diferentes.

La información relativa a la categoría sintáctica se utiliza como base tanto para el análisis sintáctico-semántico como para el morfológico. En un punto siguiente se tratará su uso en cada nivel en la gramática de unificación.

5.4.3 Adición de vocabulario

En el desarrollo de Ares hemos tomado como punto de partida conseguir obtener el léxico del sistema de forma automática a partir de la información existente en WordNet. Sin embargo, no todas las palabras que se utilizan en las descripciones procesadas se encuentran en WordNet. En concreto, en las descripciones se utilizan 641 términos, 552 de los cuales son los presentes en WordNet. Para los términos no presentes, de acuerdo con nuestro planteamiento inicial del sistema, no hemos incluido información en el léxico de forma manual.

Los 89 términos que no aparecen en WordNet son excesivamente especializados. Algunos ejemplos son "checksum", "hexadecimal", "ip", "lineprinter" y "hostname". Conviene señalar que dentro de esta cifra incluimos también las apariciones de otras órdenes dentro de las descripciones (e.g. "telnet", "textedit"). Por otra parte, es importante hacer mención del importante número de tecnicismos que se utilizan en las descripciones (e.g. "ascii", "bit", "c", "database") que sí se encuentran en WordNet, con los significados específicos de nuestro dominio (e.g. "c", con seis significados, y el 03405966 -- "a general purpose programming language closely associated with the UNIX operating system").

Para cinco términos, representativos de algunos fenómenos característicos de nuestro dominio, hemos hecho una excepción y hemos insertado en el léxico su información de forma semiautomática, a modo de ilustración de cómo podría automatizarse el proceso en algunos casos. Dos términos, "keyword" y "pathname",

ilustran el hecho del uso de términos en el manual que se forman por la composición de dos palabras (e.g. "keyword" = "key" + "word"). Para este caso, los ítems que hemos insertado en el léxico los hemos formado por la suma de los significados de los términos que los forman. Un caso sensiblemente diferente es el de "overwrite", que le hemos asociado la unión de los significados de "write" y "remove". También hemos incluido en el léxico un término formado por un verbo más preposición a modo de ejemplo de este hecho, en concreto para "look for", que le hemos asignado el significado 00536270, que comparte con "seek" y "search" (esta información sí se encuentra en WordNet, pero no hemos considerado este tipo de términos en nuestro preprocesador). La representación de este tipo de términos se puede incluir en el léxico en la forma siguiente, en concreto para el caso citado:

```
word(['look','for'],verb,[00536270]).
```

El tratamiento más detallado de estos casos particulares no se encuentra dentro de nuestros objetivos actuales. De acuerdo con la aproximación seguida, para los 84 términos restantes no hemos incluido ninguna información. El léxico obtenido de esta forma ha sido el utilizado en los experimentos que presentamos al final de este capítulo. En el apéndice final puede encontrarse el léxico completo utilizado en el sistema.

5.5 El lenguaje de representación de las descripciones

El lenguaje de representación de los documentos es un aspecto fundamental en los sistemas de recuperación de información, y en todos los sistemas que siguen la aproximación semántica está muy relacionado con los lenguajes de representación del significado (MRL) propuestos en el ámbito del PLN (e.g. scripts en el caso de FERRET).

La definición del lenguaje de representación de Ares es uno de los elementos más determinantes de todo el sistema. En nuestro caso, para definir el lenguaje de representación hemos utilizado como base tanto criterios generales de PLN, como criterios específicos del dominio. Además hemos subordinado el procesamiento de las consultas del usuario al procesamiento de las descripciones. El lenguaje de representación de las descripciones nos ha servido como base para la definición del lenguaje de representación de las consultas del usuario, debido a la proximidad existente entre el lenguaje utilizado en consultas y descripciones.

Fundamentalmente, el lenguaje de representación debe tener el suficiente poder descriptivo como para permitir mejorar el proceso de recuperación. Por otra parte, no tiene por qué incluir aspectos o detalles, que no sean relevantes para la recuperación de información en nuestro dominio.

En los siguientes apartados especificamos los requisitos, o aspectos deseables, que debe satisfacer el lenguaje de representación en un sistema como Ares; las restricciones, o peculiaridades, del dominio más importantes que hemos tenido presentes en la definición del lenguaje de representación; y por último, definimos el lenguaje de representación utilizado en Ares.

5.5.1 Requisitos del lenguaje de representación

Los siguientes, son los requisitos que se han tenido presentes, principalmente, para la definición del lenguaje de representación de las descripciones en Ares.

1. *Poder descriptivo*: El lenguaje de representación debe tener la suficiente expresividad como para permitir mejoras en la efectividad del proceso de recuperación. El lenguaje de representación debe proporcionar una expresividad y poder descriptivo mayor que el de los vectores de términos con pesos utilizados en los sistemas basados en el modelo vectorial. El lenguaje de representación debe tener una capacidad de representación del contenido de las descripciones semejante al de los sistemas de RI basados en la semántica, tales como el utilizado en FERRET basado en estructuras de tipo frame.

2. *Representación orientada a la semántica*: El lenguaje se debe centrar en la representación del significado de las descripciones. El lenguaje de representación debe ser tal que descripciones diferentes, pero con contenidos, o significados, semejantes sean representadas de forma igual o parecida. Así mismo, debe ser tal que descripciones superficialmente parecidas, pero con contenidos, o significados, diferentes, se representen de forma distinta.

3. *Disminución de la complejidad del analizador*: El lenguaje de representación debe ser tal que el proceso de análisis de los documentos se vea facilitado en la mayor medida posible (y por tanto el analizador), siempre que la efectividad del proceso de recuperación no se vea afectada. No se deben incluir atributos en el lenguaje de representación que conlleven una obtención de sus valores por el analizador de elevada complejidad, mientras no sea necesario. Como caso concreto, una representación con un nivel de detalle análogo al de los sistemas de comprensión de texto como BORIS, con una organización basada en MOPs, resulta excesiva y conlleva el desarrollo de un analizador de elevada complejidad.

4. *Orientación al cálculo de la similitud con las consultas*: En la definición del lenguaje de representación de las descripciones se debe tener presente su finalidad: permitir el cálculo de la similitud de las descripciones con las consultas del usuario. El lenguaje de representación de las descripciones debe ser tal que se representen los aspectos relevantes a las necesidades del usuario y se vea facilitado el cálculo de la similitud entre descripciones y consultas.

5. *Potenciación del uso de información léxica*: Este aspecto se encuentra de acuerdo con la tendencia actuales de los sistemas de PLN a potenciar este elemento del sistema y con conseguir la máxima explotación posible de la información obtenida de una base de datos léxica ya existente, WordNet. En primer lugar, el lenguaje de representación debe ser tal que permita la inclusión de la mayor información léxica posible, como los identificadores de los conceptos incluidos en el léxico, que pueden utilizarse para representar de forma no ambigüa el significado concreto de los términos en las descripciones. Por otra parte, el lenguaje de representación debe facilitar el desarrollo de un

analizador que se base lo máximo posible en la información existente en el léxico del sistema relativa a los conceptos y categorías sintácticas asociadas a los términos.

A estos requisitos les podemos añadir el de "adecuación al dominio", quizá uno de los más relevantes en nuestra aproximación. Esto lo tratamos en el siguiente punto.

5.5.2 Restricciones del dominio

En la definición del lenguaje de representación adecuado al dominio, se han tenido presentes dos cuesitones fundamentales: el tipo de lenguaje que se utiliza en las descripciones y en las consultas, y los objetos que son descritos en ellas. Estas dos cuestiones se tratan en los siguientes cuatro puntos, correspondiente a las restricciones del dominio más relevantes para la definición del lenguaje de representación y cómo han influido en esta última.

1. *Uniformidad de los objetos descritos*: Todas las descripciones se orientan a describir un mismo tipo de objeto: una orden de UNIX. En la definición del lenguaje de representación debemos incluir elementos que resulten especialmente adecuados para la representación de este tipo de objetos.

2. *Brevedad y concisión de las descripciones*: Las descripciones de las órdenes se encuentran formadas en su totalidad por una o dos oraciones, y en su mayoría por una sola. Además, la información que se presenta en ellas es muy concisa, a diferencia de los dominios tratados en otros sistemas de recuperación de información basados en la semántica, como FERRET o NLDB, que siguen una aproximación parecida a los sistemas de extracción de información basada en "text skimming". En nuestro caso, el lenguaje de representación debe permitir representar la práctica totalidad de la información existente en las descripciones. No se puede seguir una aproximación análoga a la de los sistemas de text skimming, en los que se selecciona sólo una parte de la información existente en el texto [Hobbs92].

3. *Uso del lenguaje enunciativo*: Todas las descripciones a procesar se encuentran formadas por frases enunciativas, de forma tal que no es necesario incluir en el lenguaje de representación elementos que den cuenta de otros tipos de oraciones, como las interrogativas, ni se hace uso de cuantificadores (como el existencial o el universal). Tampoco se utilizan verbos en diferentes tiempos, todas las expresiones se encuentran formadas por verbos en presente. Todos estos aspectos, que se suelen incluir en sistemas de PLN como los interfaces en LN a bases de datos, no son necesarios en nuestro dominio [Carrol93, Allen94, Ritchie95].

4. *Similitud entre consultas y descripciones*: Los usuarios expresan su necesidad mediante oraciones enunciativas, al igual que en Argos. Las consultas de los usuarios se formulan en un lenguaje muy parecido al utilizado en las descripciones. En Ares utilizamos el mismo lenguaje de representación de las descripciones para representar las consultas de los usuarios. La similitud

```
frames          --> []; frame, frames.
frame           --> slotList.
slotList        --> []; slot, slotList.
slot            --> role, fillerList
role            --> [action]; [comparison];
                    [destination]; [duration]; [instrument];
                    [manner]; [object]; [purpose]; [source];
                    [time].
fillerList      --> filler; filler, fillerList.
filler          --> [head], head, [modif], modifList.
head            --> termconcept.
modifList       --> []; wordRepr, modifList.
termconcept     --> [termconcept], term, conceptList.
conceptList     --> concept; concept, conceptList.
term            --> atom.
concept         --> integer.
```

Figura 5.8: Formalismo para la representación de las descripciones.

existente entre el lenguaje de las consultas y el de las descripciones puede interpretarse como un supuesto, o como una imposición a los usuarios. En cualquier caso, existen estudios experimentales que avalan la suposición de que la mayoría de las consultas de los usuarios se hacen de esta forma, mediante oraciones sencillas [Guindon88]. En nuestro dominio, seguir esta aproximación nos ha permitido procesar el 83% de las consultas que utilizamos en los experimentos que presentamos al final del capítulo.

5.5.3 Definición del lenguaje de representación

En Ares, el significado de las descripciones y consultas se representa mediante estructuras de tipo *frame*, como la utilizada en sistemas como FERRET [Mauldin91] y en sistemas de PLN [Gazdar89, Allen94]. La definición de la estructura de los frames y la de los slots que los integran se basa en los conceptos de role temático y caso. Utilizamos ambos conceptos en una forma próxima a la habitual en sistemas de PLN [Allen94], ya tratada en un capítulo anterior, y próxima también a como se realiza en el sistema UCF [Wendlandt91], en el que, como se ha tratado en un capítulo anterior, se potencia la utilización de información léxica. De forma análoga a como se ha realizado en este último sistema, el conjunto de casos que utilizamos lo hemos definido basándonos en los originales introducidos por Fillmore [Bruce87, Allen94], adaptándolos a nuestro dominio.

Una representación basada en frames de casos resulta especialmente adecuada para la representación de la semántica de expresiones como las que forman las descripciones, con características ya señaladas en el apartado anterior. Por otra parte, en sistemas orientados a la recuperación de componentes software, como el de Wood y Sommerville [Wood88], se subraya la adecuación de este tipo de representación para las órdenes de UNIX: objetos que básicamente se encuentran orientados a realizar una función (o acción) sobre otros objetos que juegan determinados roles temáticos (fuente, destino). De esta forma, la definición de una representación basada en roles temáticos, o casos, se adapta tanto al lenguaje

- action: la acción que se realiza en la oración, correspondiente al verbo
- object: la entidad que es afectada por la acción
- comparison: la entidad comparada con el objeto
- destination: el destino del objeto
- duration: el tiempo a lo largo del cual transcurre la acción
- instrument: el objeto utilizado para realizar la acción
- manner: la forma en que se realiza una acción
- purpose: el propósito de la acción
- source: el lugar donde se realiza la acción o de donde parte el objeto
- time: el momento en que se realiza, o comienza a realizarse, la acción

Figura 5.9: slots utilizados en la representación basados en roles temáticos.

utilizado en las descripciones como a los objetos que se describen en ellas, las órdenes de UNIX.

En la figura 5.8 se incluye la definición del formalismo utilizado para la representación de las descripciones. Utilizamos DCG (que en este caso puede considerarse como alternativa al habitual BNF) como formalismo para la definición de la gramática que especifica el lenguaje de representación, el mismo que hemos utilizado en el desarrollo del analizador y que trataremos en un punto siguiente. La representación de una descripción esta formada por una lista de frames, de forma tal que cada frame se corresponderá con la representación semántica de cada oración incluida en la descripción. Una representación incluirá dos frames como máximo, por la propia naturaleza de las descripciones.

Cada frame está compuesto por una serie de slots. El nombre de los slots viene dado por el conjunto de roles temáticos. En la definición de nuestra representación hemos incluido nueve roles temáticos que permiten representar los principales papeles que pueden jugar los objetos que intervienen en una acción, además de el correspondiente a la acción misma. El role temático *agent* no se incluye: en todos los casos el agente de la acción es el nombre de la misma orden que se describe y no es interesante para nuestro objetivo del cálculo de la similitud.

El significado que representa cada uno de estos roles es el utilizado habitualmente en sistemas de casos [Bruce87, Allen94] y se describe en la figura 5.9. El valor asociado a cada role temático representa la entidad que juega este papel en la descripción (salvo la acción en sí misma). Así el valor de cada role viene dado por una estructura apropiada para descripbir los objetos que intervienen en una órden de UNIX y está formada esencialmente por dos partes: una cabeza y una lista de modificadores. La cabeza tiene como valor el del concepto que representa a la entidad en sí misma. Los modificadores, los conceptos que se corresponden con las palabras que sirven como modificadores de la cabeza en la oración y que definen con más precisión la entidad que ocupa ese role.

Las unidades más sencillas incluidas en la representación son asociaciones término-lista de conceptos, que denominaremos a partir de ahora asociaciones *término-concepto*. En los término-conceptos, se incluyen los conceptos existentes en el léxico del sistema correspondientes al significado con que intervienen las palabras en las oraciones. Se incluye también la representación los términos correspondientes a las palabras en una forma canónica, independiente de variaciones morfológicas.

El siguiente es un ejemplo de representación de una descripción, la correspondiente a la orden "cd" para cambiar de directorio de trabajo:

cd: "change working directory"

```
[[[action,
     [[head(
         [termconcept(change,[50777,49698])]),
       modif([])]]],
  [object,
     [[head(
          [termconcept(directory, [3214387])]),
       modif(
          [termconcept(working, [25152])])]]]]]]).
```

El frame esta formado en este caso por un slot correspondiente a la acción, *action* y otro correspondiente al role temático *object*, el objeto sobre el que se realiza la acción. Los conceptos que aparecen en la representación son los correspondientes a los significados con que se utilizan las palabras en la descripción. En el ejemplo anterior, la palabra "change" se representa por dos significados de los catorce que tiene en WordNet (seis como verbo y ocho como nombre). Estos significados se corresponden, entre los seis que tiene como verbo, con los subrayados:

```
0901469:    exchange, chage, interchange -- (transfer; cause to
            change ownership)
0068165:    change, dress -- (get dressed; "change before you go
            to the opera")
0066589:    change, exchange, convert -- ("as of currency, for
            example")
0050777:    change, alter -- (cause to change; make different)
0049698:    change, alter, vary
0046130:    change, become_different -- ("the weather changed last
            night")
```

El objeto que se encuentra asociado al role temático *object* se encuentra representado por el significado 3214387 de la palabra "directory". La palabra "working" no es pasada a su forma "work" por encontrarse en el léxico como adjetivo en la primera forma, y actúa como modificador con el significado 25152.

De acuerdo con nuestros objetivos, la representación semántica de expresiones en las que aparecen términos diferentes, pero con significados equivalentes, es la misma o muy próxima. Así la expresión

"alter the active directory"

tiene la misma representación semántica que la anteriormente comentada. El término "alter" se encuentra representado por los significados 050777 y 049698, comunes con "change". El término "active" tiene asociado el 25152, el mismo que "working". La estructura de los frames de ambas expresiones sería también la misma:

```
[[[action,
     [[head(
         [termconcept(alter,[50777,49698])]),
       modif([])]]],
  [object,
     [[head(
          [termconcept(directory, [3214387])]),
       modif(
          [termconcept(active, [25152])])]]]]]]]).
```

El siguiente ejemplo corresponde a la representación de la descripción de la orden *lprm*, para eliminar trabajos de la cola de impresión.

lprm: "remove jobs from the printer queue"

```
[[[action,
     [[head(
         [termconcept(remove, [951139, 70485])]),
       modif([])]]],
  [object,
     [[head(
          [termconcept(job, [2109166])]),
       modif([])]]],
  [source,
    [[head(
         [termconcept(queue, [3233731])]),
      modif([termconcept(printer, [2291438])])]]]]]
```

En este caso la descripción incluye un slot para la acción, *action*, y otros para los roles temáticos *theme* y *source*.

Es importante hacer notar que nuestro lenguaje de representación, además de permitir afrontar los problemas relacionados con la sinonimia y la polisemia, proporciona un mecanismo para la mejora de *precision*. Expresiones que incluyen los mismos términos, con los mismos significados, pero que tienen un significado completo diferente, tienen representaciones semánticas diferentes. Como ejemplo, podemos considerar las siguientes expresiones:

"remove a text file"
"remove a text from a file"

Estas dos expresiones tienen significados diferentes aunque contienen los mismos términos con los mismos significados. Como es deseable, la representación semántica de cada una de las expresiones es diferente:

"remove a text file"

```
[[[action,
    [[head([termconcept(remove, [951139, 70485])]),
      modif([])]]],
  [object,
    [[head([termconcept(file, [3236400])]),
      modif(
        [termconcept(text, [3210873, 3201596])])]]]]]
```

"remove text from a file"

```
[[[action,
    [[head(
        [termconcept(remove, [951139, 70485])])],
      modif([])]]],
  [object,
    [[head(
        [termconcept(text, [3210873, 3201596])])],
      modif([])]]],
  [source,
    [[head(
        [termconcept(file, [3236400])])],
      modif([])]]]]]
```

De esta forma, las representaciones semanticas permiten su diferenciación y proporcionan la base para la mejora de la *precision*. En el siguiente punto se tratan aspectos adicionales de la representación de las órdenes utilizada en Ares, al tratar la definición de la gramática de unificación, con la que se encuentra fuertemente interrelacionada.

5.6 La gramática de unificación

En la gramática de unificación hemos incluido toda la información lingüística, adicional al léxico, que se utiliza en Ares. En la concepción del sistema, como hemos señalado anteriormente, nos hemos planteado establecer una clara división entre las dos componentes fundamentales que permiten el procesamiento de las descripciones y las consultas. En el léxico se incluye información obtenida automáticamente de WordNet que en general no es específica del dominio. En contraste con el léxico, la gramática de unificación se ha desarrollado de forma enteramente manual y es en ella en donde se incluye la información lingüística específica del dominio. Esta división se puede observar en la organización del sistema presentada en la figura 5.6.

La gramática de unificación permite realizar a Ares un análisis morfológico, sintáctico y semántico de las expresiones, orientado a obtener su representación semántica en el formalismo presentado en el punto anterior. La gramática se encuentra dividida fundamentalmente en dos componentes: una orientada al procesamiento morfológico y otra orientada al análisis sintáctico-semántico. La componente morfológica es en cierta medida independiente del dominio, pero queda incluida dentro de la gramática de unificación por la necesidad de su desarrollo

manual y la posibilidad de su adaptación al dominio en posibles evoluciones del sistema. Presentamos a continuación ambas componentes.

5.6.1 Componente morfológico

El resultado del análisis morfológico realizado por Ares es en parte semejante al realizado por los algoritmos de extracción de raíces (stemming) utilizados en los sistemas de recuperación de información, o el utilizado en el módulo de RI de Argos. Esencialmente, el analizador morfológico proporciona, dada cualquier flexión de una palabra, el término correspondiente existente en el léxico. Proporciona, además, sus posibles categorías sintácticas. El análisis morfológico es posible gracias a la información existente en el léxico del sistema.

Para la implementación de la componente morfológica de la gramática de unificación hemos utilizado las reglas de separación (*detachment*) incluidas en el analizador morfológico que incluye el interfaz a WordNet [Miller93]. Estas 18 reglas aparecen en la figura 5.10. Si una palabra termina en un sufijo de los existentes en la segunda columna, se sustituye por el correspondiente de la tercera columna. Estas transformaciones de las terminaciones de las palabras son posibles para las categorías sintácticas asociadas, especificadas en la primera columna. De esta forma, dada una palabra en su forma original, tal como se encuentra en una expresión de entrada, se le aplican las reglas de separación de sufijos hasta comprobar si se encuentra en su forma canónica en el léxico del sistema, y con la categoría sintáctica permitida para la regla de separación aplicada. Así, por ejemplo, para una palabra como "switching", el resultado del procesamiento morfológico sería el término "switch", y la categoría sintáctica *verb*. Para "switches", se obtiene igualmente el término "switch", pero con dos posibles categorías sintácticas, *noun* y *verb*.

Para la implementación del analizador morfológico, hemos definido un conjunto de reglas en DCG que se corresponden una a una con las reglas de la figura 5.10. En ellas se incluye el sufijo flexivo, la parte de la palabra sin sufijo -básicamente la raíz- y el sufijo correspondiente para la transformación a la forma en que aparece en el léxico. Se incluye además, como primer argumento, la categoría sintáctica para la que es válida la aplicación de la regla. De acuerdo con esto, las reglas en DCG correspondientes a las 5 primeras que aparecen en la figura 5.10 son:

```
morf(noun, R, []) --> root(R), [s].
morf(noun, R, [s]) --> root(R), [s,e,s].
morf(noun, R, [x]) --> root(R), [x,e,s].
morf(noun, R, [z]) --> root(R), [z,e,s].
morf(noun, R, [c,h]) --> root(R), [c,h,e,s].
```

Las restantes se pueden encontrar en el apéndice final. De estas reglas se hace uso desde el predicado de acceso al analizador, *morphie*. Este predicado instancia las variables lógicas que tiene como argumentos a la palabra en cualquiera de sus formas flexivas, el término existente en el léxico del sistema y su categoría sintáctica.

Noun	"s"	""
Noun	"ses"	"s"
Noun	"xes"	"x"
Noun	"zes"	"z"
Noun	"ches"	"ch"
Noun	"shes"	"sh"
Verb	"s"	""
Verb	"ies"	"y"
Verb	"es"	"e"
Verb	"es"	""
Verb	"ed"	"e"
Verb	"ed"	""
Verb	"ing"	"e"
Verb	"ing"	""
Adj	"er"	""
Adj	"est"	""
Adj	"er"	"e"
Adj	"est"	"e"

Figura 5.10: Reglas morfológicas de separación de sufijos.

Los siguientes son ejemplos de consultas, desde el intérprete de órdenes, de accesos al predicado *morphie*, correspondientes a los casos de "switching" y "switches" antes tratados:

```
?- morphie(switching, Termino, Categoria).
    Termino = switch,
    Categoria = verb ;
    no
?- morphie(switches, Termino, Categoria).
    Termino = switch,
    Categoria = noun ;
    Termino = switch,
    Categoria = verb ;
    no
?-
```

Las reglas de separación utilizadas son válidas para un número importante de términos del Inglés y pueden complementarse con una lista de excepciones [Miller93]. En nuestro caso, no ha sido necesario incluir esta información en el sistema: el analizador morfológico implementado cubre el 100% de las palabras utilizadas en las descripciones y en las consultas.

Como ya hemos señalado, el análisis morfológico tiene como propósito fundamental la identificación de las posibles categorías sintácticas de una palabra y la obtención del término correspondiente en el léxico. La transformación de la palabra al término tiene un propósito esencialmente análogo al de los algoritmos de extracción de raíces utilizados en RI tratados en capítulos anteriores: la obtención de una forma canónica de las palabras. Sin embargo, a diferencia de los algoritmos de stemming, el analizador morfológico no proporciona la raíz de la palabra, sino su forma canónica, tal como se incluye en el léxico. Además, es más preciso y no aplica reglas

```
?- sentences(Frame,['execute','a','command','or','script','at','a',
                    'specified','time'],[]).
   Frame = [[[action,
              [[head([termconcept(execute,
                       [1016285, 1015621, 984018, 983489, 691099, 662251,
                        397674])])],
                modif([])]]],
             [object,
              [[head([termconcept(command,
                       [6139920, 3506089, 3266508, 2965262, 2584813])])],
                modif([])],
               [head([termconcept(script,
                       [3448992, 3206082, 3189912])])],
                 modif([])]]],
             [time,
              [[head([termconcept(time,
                       [6684977, 6675222, 6617340, 6614763, 6610265,
                        3560779, 3551862, 2677621, 12418])])],
                modif([termconcept(specified,
                       [546350])])]]]]]

?- sentences(Frame,['display','a','string','in','large','letters'],[]).
   Frame = [[[action,
              [[head([termconcept(display,
                       [1004577, 855169])])],
                modif([])]]],
             [object,
              [[head([termconcept(string,
                       [6174696, 3450292, 2440981, 1891440, 1890788,
                        1750340, 1709685, 1709563])])],
                modif([])]]],
             [manner,
              [[head([termconcept(letter,
                       [3370596, 3319310, 3285514])])],
                modif([termconcept(large,
                       [696979, 691740, 766564, 733637, 554188])])]]]]]]
```

Figura 5.11: Análisis de descripciones.

de forma incorrecta -como puede ocurrir con los algoritmos de extracción de raíces con los posibles errores de infrarradicación y sobrerradicación [Frakes92]- al comprobar la existencia en el léxico del término obtenido.

5.6.2 Componente sintáctico-semántico

El analizador sintáctico-semántico tiene como entrada la lista de símbolos correspondientes a una descripción y como salida, la representación de la descripción en el lenguaje de representación descrito en el punto 5.5. El predicado de acceso al analizador es *sentences*. En la figura 5.11 aparecen dos ejemplos de análisis obtenidos para las descripciones de las órdenes *at* y *banner*, respectivamente.

En el analizador no existe una clara separación de un componente sintáctico y otro semántico, sino que ambos tipos de información se encuentran integrados

simultáneamente en las reglas de la gramática, que tiene como objetivo fundamental la obtención de la representación semántica de las expresiones de entrada, siguiendo una aproximación semejante a la presentada en [Gazdar89].

El desarrollo de la gramática se ha realizado de forma incremental, es decir, incluyendo nuevas reglas o modificaciones sobre un conjunto básico de ellas, mediante sucesivas comprobaciones de los análisis obtenidos para las descripciones. De acuerdo con esto, hemos partido de un conjunto de reglas inicial que permitía realizar el análisis de un conjunto de descripciones relativamente reducido y mediante la inclusión de nuevas reglas hemos llegado a conseguir analizar correctamente un 64.8% de las descripciones del manual (280 de 432). Sistemas que siguen una aproximación basada en la semántica consiguen coberturas análogas o inferiores (e.g. el sistema FERRET [Mauldin91] analiza entre el 15% y el 30% de las oraciones de los textos, adecuándose a su estrategia de *text skimming*). En el caso de Ares conseguir esta cobertura ha resultado suficiente para mejorar la efectividad del proceso de recuperación, como veremos en un punto siguiente.

Disponer de los diferentes elementos de información (i.e. las descripciones, el léxico y la gramática, fundamentalmente) integrados en un entorno declarativo como el formado por la combinación de Prolog y DCG, ha facilitado el proceso de desarrollo de la gramática. De esta forma, resulta posible conocer de forma inmediata datos concretos acerca de las descripciones mediante consultas al intérprete de Prolog o mediante la construcción de predicados específicos. Por ejemplo, podemos obtener de forma directa todas las descripciones en las que aparece un sólo verbo, o dos, o ninguno, o conocer el hecho de que en ninguna descripción aparecen tres verbos. Este tipo de datos nos ha permitido adquirir conocimiento sobre las restricciones del lenguaje utilizado en las descripciones, que hemos incluido en forma de reglas en la gramática. También resulta inmediato conocer las descripciones que son analizadas correctamente mediante la gramática de unificación y saber cuáles no, viéndose facilitada la inferencia de las reglas que se deben incorporar en la gramática para conseguir ampliar su cobertura.

En la definición de la gramática hemos utilizado las restricciones del dominio señaladas en el punto 5.5. para la definición de la forma de representación de las descripciones, y otras adicionales más relevantes al desarrollo de la gramática, aunque todas ellas se encuentran muy interelacionadas. Las principales restricciones identificadas en el conjunto de descripciones y que se han utilizado en la definición de la gramática son las siguientes:

1. Las descripciones de los comandos están formadas por oraciones enunciativas en activa. No aparecen oraciones en pasiva. Esto conlleva que el agente de todas las acciones venga dado por el sujeto y la acción por el verbo.

2. La mayoría de las descripciones se encuentran formadas por una oración. Un cierto número de ellas por dos, y ninguna por tres o más.

3. La mayoría de las descripciones son frases verbales o sintagmas verbales (*verb phrases*). Son oraciones que presentan el sujeto elidido: es el comando que se describe.

4. En las oraciones no aparecen proposiciones subordinadas.

5. En una gran cantidad de oraciones, el objeto viene dado por la frase nominal inmediatamente posterior al verbo.

6. El role temático asociado a las frase nominales se encuentra fuertemente determinado por las preposiciones que las introducen.

Otras características adicionales de las descripciones facilitan el desarrollo de la gramática y, aunque no se representen en ella explícitamente, sí se utilizan implícitamente. Por ejemplo, no es necesario la comprobación de concordancia de género y número entre sujeto y predicado, y por ello no se incluye explícitamente este tipo de restricción en la gramática. Tampoco es necesario disponer de mecanismos que resuelvan la ambigüedad referencial que introduce el uso de pronombres porque en las descripciones no se hace uso de ellos.

Basándonos en estas consideraciones hemos construido la gramática utilizada en el sistema y que se incluye de forma completa en el apéndice final. Comentamos a continuación las relaciones existentes entre las restricciones anteriormente señaladas y las reglas de los elementos más importantes de la gramática.

La regla *sentences*, correspondiente a una descripción, se define basándonos en el hecho de que las descripciones están formadas a lo sumo por dos oraciones, con un nexo entre ambas (*coord*). El resultado del análisis es la unión de los frames correspondientes a cada oración:

```
sentences([CaseFrame]) -->
    sentence(CaseFrame).
sentences([CaseFrame1, CaseFrame2]) -->
    sentence(CaseFrame1),
    coord,
    sentence(CaseFrame2).
```

En la regla *sentence*, se tienen presentes las restricciones anteriormente señaladas relativas a la elipsis del sujeto, la obtención de la acción y la posición del objeto, así como la de las frases nominales correspondientes a los restantes roles temáticos.

```
sentence([ActionSlot,ObjectSlot|SlotList]) -->
    verbs(ActionFiller),
    noun_phrases(ObjectFiller),
    { cases(SlotList); [], { SlotList = [] } },
    { ActionSlot = [action,ActionFiller],
      ObjectSlot = [object,ObjectFiller] }.
```

La acción se obtiene a partir del verbo, o los verbos que aparecen al comienzo de la frase. La regla de formación de *verbs* es:

```
verbs( [ [head([VerbTermConcept]),modif([])] ]) -->
     verb(VerbTermConcept).
verbs( [ [head([VerbTermConcept]),modif([])] | VerbTermConceptL]) -->
     verb(VerbTermConcept),
     coord,
     verbs(VerbTermConceptL).
```

En la regla de formación *verb* incluimos la posibilidad de aparición de verbos como "copy" o "look" y verbos con preposiciones como "look for":

```
verb(termconcept(Verb,ConceptL)) -->
     [Word],
     { morphie(Word,Verb,verb),
       word([Verb],verb,ConceptL) }.
verb(termconcept(Verb,ConceptL)) -->
     [Word,Prep],
     { morphie(Word,Verb,verb),
       word([Verb,Prep],verb,ConceptL) }.
```

En la regla *sentence*, los valores de los slots correspondientes a los roles temáticos, o casos, adicionales al objeto se obtienen a partir de las frases nominales (*cases*) que aparecen tras el verbo y las frases nominales correspondientes al objeto. De acuerdo con las restricciones anteriormente señaladas, el role temático se encuentra fuertemente determinado por la preposición que introduce la frase nominal correspondiente. Esta información es utilizada en la regla *cases*:

```
cases([Slot]) -->
     case(Slot).
cases([Slot|SlotList]) -->
     case(Slot),
     cases(SlotList).
case([Role,Filler]) -->
     case_marker(RoleList),
     noun_phrases(Filler),
     { member(Role,RoleList) }.
```

Las preposiciones pueden utilizarse como base para la obtención de información semántica, al igual que en el sistema UCF [Wendlant91]. En la gramática se incluye información relativa a 15 preposiciones que se utilizan como indicadores de los roles temáticos, de la forma habitual en los analizadores de casos [Allen94]. Los roles temáticos asociados a las proposiciones se incluyen en la gramática en forma de hechos Prolog. Como ejemplo, para las partículas "to", "by", "for", "in", se incluyen los hechos:

```
prepcv(to,[destination,purpose]).
prepcv(by,[instrument,manner]).
prepcv(for,[purpose,duration]).
prepcv(in,[source,destination,manner,time]).
```

En el primero de los hechos se representa el que la preposición "to" se utiliza como indicadora de los roles temáticos *destination* y *purpose*.

La uniformidad de la estructura de las frases nominales de nuestro dominio es utilizada para la obtención del término que se incluye como cabeza (*head*) en los valores de los slots en la representación y los términos que actúan como sus modificadores (*modif*) en las representaciones. Las palabras correspondientes a la parte *head* de la representación se obtienen de la parte central de las frases nominales. Los modificadores se obtienen a partir de los calificadores (*qualifiers*) que preceden al núcleo y de los complementos (*complement*) que le siguen. Esto se tiene presente en la definición de la regla *noun_phrases*.

```
noun_phrases([NounPhraseAn]) -->
     noun_phrase(NounPhraseAn).
noun_phrases([NounPhraseAn|NounPhraseAnL]) -->
     noun_phrase(NounPhraseAn),
     coord,
     noun_phrases(NounPhraseAnL).

noun_phrase([head(HeadList),modif(ModifList)]) -->
     ( det;
       []),
     ( qualifiers(ModifList1);
       [],
       { ModifList1 = []}
     ),
     np_head(HeadList),
     ( complement(ModifList2);
       [],
       { ModifList2 = []} ),
     { append(ModifList1,ModifList2,ModifList) }.
```

La parte de la gramática correspondiente al componente sintáctico semántico, consta de 25 reglas correspondientes a 15 símbolos no terminales y tiene como símbolos terminales preposiciones, artículos, conjunciones, y las palabras que pueden obtenerse mediante el analizador morfológico, *morphie*, tratado anteriormente, y que a su vez se basa en los 641 términos existentes en el léxico.

Las reglas incluidas en la gramática son ambiguas, en el sentido de que existen determinadas descripciones que tienen varios análisis, o representaciones posibles. Así por ejemplo, para la descripción de la orden "at", presentada anteriormente en la figura 5.11. se obtienen realmente dos análisis, uno de los cuales es incorrecto y no se ha incluido en esta figura. Eliminar estos análisis incorrectos, o hacer menos ambigua la gramática resultaría factible mediante la adición de reglas más restrictivas, como por ejemplo reglas correspondientes a restricciones selectivas (*selectional restrictions*), en la forma habitual en sistemas de PLN para análisis parecidos [Allen94]. En Ares no hemos incluido esta información porque no necesitamos obtener representaciones únicas de las descripciones para nuestro objetivo: mejorar la efectividad del proceso de recuperación. El problema que plantean los análisis múltiples se resuelve en la definición del método del cálculo de la similitud, que tratamos en un punto siguiente. Siguiendo esta aproximación, conseguimos reducir el esfuerzo de desarrollo de la gramática.

De acuerdo con la organización de Ares, la gramática de unificación desarrollada se utiliza para el análisis de las descripciones y de las consultas de los usuarios. Al

igual que para la definición del formalismo de representación, en el desarrollo de la gramática hemos utilizado solamente como base las descripciones de las órdenes existentes en el manual. Esta aproximación a permitido que la gramática desarrollada analice el 86.4% de las consultas que aparecen en el conjunto utilizado para los experimentos centrados en la efectividad que presentamos en un punto posterior de este capítulo.

5.7 El cálculo de la similitud

El valor de la similitud entre las consultas y las descripciones se utiliza como base para la ordenación por relevancia de las órdenes a una consulta del usuario. El cálculo de la similitud se basa en el lenguaje de representación de las órdenes y las consultas.

Como es lógico, la función de similitud se encuentra orientada a que consultas y descripciones que tengan un mismo significado, tengan un valor alto de similitud. En la definición de la función de similitud se potencia el uso de la información léxica que se encuentra incluida en la representación de descripciones y consultas. Desde un punto de vista general, la función de similitud utiliza los identificadores de conceptos incluidos en las representaciones como base para la mejora del *recall* y la estructura semántica de las descripciones como medio para la mejora de la *precision*. De esta forma, representaciones que incluyen los mismos conceptos, pero presentan una estructura semántica diferente tienen un valor bajo de la similitud.

En la definición de la función de similitud se integran criterios introducidos en aproximaciones basadas en frecuencias de aparición de términos, como la del modelo del espacio vectorial, con criterios que se centran en las peculiaridades de la utilización de una representación semántica, como en los sistemas UCF y FERRET, tratados en el capítulo anterior.

Definimos a continuación las funciones utilizadas para el cálculo de la similitud entre frames, correspondientes a representaciones de descripciones y consultas. Puntualizamos algunos aspectos y señalamos algunas alternativas a las definiciones. En la implementación del cálculo de la similitud en Ares hemos utilizado las que se especifican, y con ellas se han realizado los experimentos que se presentan en un punto siguiente.

- *Similitud entre frames*: La similitud entre dos estructuras de tipo frame es la suma de los valores de las similitudes entre sus slots comunes o compatibles, entendiendo por slots compatibles aquellos que tienen el mismo nombre o role temático.

$$simFrame(f_a, f_b) = \sum_{\forall ieRole} simSlot(sl_{f_a i}, sl_{f_b i})$$

  En la definición de la similitud entre frames se puede incluir de forma sencilla la utilización de pesos diferentes para cada slot o role temático, como

en el sistema UCF. Sin embargo, en nuestro sistema todos los roles temáticos tienen el mismo peso.

La mayoría de las descripciones de las órdenes se encuentran representadas por un sólo frame, así como todas las consultas de los usuarios de los experimentos que se presentan en un punto siguiente. De esta forma, la similitud entre frames proporciona directamente el valor de la similitud entre consulta y descripción. Para los casos en que la descripción es representada por dos frames, el valor de la similitud entre consulta y descripción es el máximo de los de la similitud de cada uno de los frames de la descripción con el de la consulta.

- *Similitud entre slots compatibles*: La similitud entre dos slots con el mismo role temático se define como la suma de todas las similitudes entre los fillers de sus listas de fillers correspondientes:

$$simSlot(sl_a, sl_b) = \sum_{i=1}^{n_{sl_a}} \sum_{j=1}^{n_{sl_b}} simFiller(fl_{sl_a i}, fl_{sl_b j})$$

en donde n_{sla} y n_{slb} son el número de elementos que tienen las listas de fillers asociadas a los slots sl_a y sl_b, respectivamente.

- *Similitud entre fillers*: La similitud entre dos fillers se define como la suma de las similitudes entre sus cabezas y sus modificadores:

$$simFiller(fl_a, fl_b) = simHead(h_{fl_a}, h_{fl_b}) + simMods(m_{fl_a}, m_{fl_b})$$

- *Similitud entre núcleos*: La similitud entre núcleos incluidos en fillers de slots se define como el valor máximo de las similitudes entre las asociaciones término-concepto incluidas en cada uno de ellos.

$$simHead(h_a, h_b) = max_{\forall i,j}\left(simTermConcept(tc_{h_a i}, tc_{h_b j})\right)$$

- *Similitud entre modificadores*: La similitud entre modificadores incluidos en fillers de slots se define como el valor máximo de las similitudes entre las asociaciones término-concepto incluidas en cada uno de ellos.

$$simMods(m_a, m_b) = max_{\forall i,j}\left(simTermConcept(tc_{m_a i}, tc_{m_b j})\right)$$

En la definición de la similitud entre núcleos y en la de modificadores, tomamos en ambos casos el máximo de las similitudes entre las asociaciones término-concepto. Ya que el número de asociaciones término-concepto incluidas en los modificadores es mayor que en el núcleo, en la mayoría de los casos, conseguimos de esta forma dar un mayor peso a las asociaciones término-concepto del núcleo. Esto resulta adecuado a que los términos del núcleo juegan un papel más importante en las descripciones.

- *Similitud entre asociaciones término-concepto*: La similitud entre dos asociaciones término-concepto es nula cuando la intersección de los conjuntos de conceptos correspondientes es vacía, es decir, cuando no tienen un significado común. En caso de tener una intersección no vacía, o tener un significado común (o varios), el valor de la similitud es el mínimo de los pesos de los términos, calculados a partir de su frecuencia de aparición en los documentos.

$$simTermConcept(tc_a, tc_b) = w_{t_a} \cdot w_{t_b} \cdot \theta(tc_a, tc_b)$$

$$w_t = log_2(n / df_t)$$

$$\theta(tc_a, tc_b) = \begin{cases} 0 \text{ si } conc(tc_a) \cap conc(tc_b) = \varnothing \\ 1 \text{ en caso contrario} \end{cases}$$

Siendo n el número de documentos existentes en la colección y df_t el número de documentos en que aparece el término t.

Una alternativa a los pesos utilizados en la definición de esta función, basados en la frecuencia de aparición de los términos, sería la utilización de pesos basados en la frecuencia de aparición de los conceptos, como se hace en [Voorhees93]. En Ares hemos utilizado en una primera aproximación la anterior expresión, que facilita el desarrollo e interpretación de los experimentos que presentamos en el siguiente punto.

En suma, las definiciones introducidas para el cálculo de la similitud, se encuentran dirigidas a conseguir, de acuerdo con la concepción inicial de Ares, implementar simultáneamente un mecanismo orientado a la mejora del *recall*, basado en la información léxica de propósito general obtenida de WordNet, y un mecanismo orientado a la mejora de la *precision*, basado en la información específica del dominio utilizada para la definición del formalismo de representación de descripciones y consultas, y la gramática de unificación. En la definición de la función de similitud se utiliza la estructura semántica de las representaciones, de forma tal que descripciones que son superficialmente semejantes pero que tienen significados diferentes (y una estructura semántica diferente), tienen un valor bajo de similitud.

Consideraremos a continuación tres expresiones sencillas para ilustrar las definiciones introducidas en el cálculo de la similitud:

e_1: "remove files or directories" (*rm*)
e_2: "delete a text file"
e_3: "remove text from a file"

La primera de ellas se corresponde con la descripción corta de la orden *rm*. Las dos siguientes son semejantes a las de un ejemplo comentado anteriormente y describen dos operaciones muy diferentes. Utilizaremos estas expresiones para ilustrar dos aspectos principales de la definición de la función de similitud:

a) expresiones diferentes superficialmente, con diferentes términos, pero con significados semejantes, tienen un valor de similitud alto.

b) expresiones iguales superficialmente, con términos iguales, pero con significados diferentes, tienen un valor bajo de similitud.

Las expresiones e_1 y e_2 son un caso representativo de a), en el que las expresiones sólo comparten un término ("file"), pero describen dos operaciones que resulta deseable que tengan un valor de similitud alto.

Las expresiones e_1 y e_3 son un caso representativo de b), en el que las expresiones tienen un mayor número de términos en común que en el caso anterior, pero sin embargo representan operaciones semánticamente diferentes en mayor medida que las anteriores, y resulta deseable que tengan un valor de similitud más bajo.

Como veremos a continuación, la similitud que se obtiene para los frames correspondientes a estas expresiones es, de acuerdo con lo deseado:

$$simFrame(f_1, f_2) = 46.4911$$
$$simFrame(f_1, f_3) = 23.1111$$

En la figura 5.12. se presentan los análisis proporcionados por el analizador para las tres expresiones. Como puede verse, las representaciones correspondientes a e_1 y e_2 tienen estructuras semánticas parecidas en las que los núcleos de los slots *action* y *object* tienen significados comunes. Sin embargo, las representaciones correspondientes a e_1 y e_2 tienen estructuras semánticas diferentes en las que sólo el slot *action* tiene significados comunes. Detallamos a continuación, a modo de ilustración, el cálculo de estas similitudes.

a) Cálculo de la similitud entre las expresiones e_1 y e_2.

La similitud entre los frames f_1 y f_2 que representan las expresiones e_1 y e_2 es:

$$simFrame(f_1, f_2) = simSlot(sl_{f_1 action}, sl_{f_2 action}) + simSlot(sl_{f_1 object}, sl_{f_2 object})$$

Para la similitud de los slots correspondientes a las acciones, ya que en este caso cada slot tiene un único filler, y ninguno de los fillers tiene modificadores:

$$simSlot(sl_{f_1 action}, sl_{f_2 action}) = simFiller(fl_{f_1 action1}, fl_{f_2 action1})$$
$$simFiller(fl_{f_1 action1}, fl_{f_2 action1}) = simHead(h_{f_1 action1}, h_{f_2 action1})$$

Con la parte *head* de los fillers:

$$h_{s_{f1action}} = head([termconcept(remove,[983048,951139,951020,179903,70485])])$$
$$h_{s_{f2action}} = head([termconcept(delete,[633190,398344,70485])])$$

e_1: "remove files or directories" (*rm*)

```
Frame1 = [[[action,
                [[head([termconcept(remove,
                          [983048, 951139, 951020, 179903, 70485])]),
                  modif([])]]],
               [object,
                [[head(
                     [termconcept(file,
                         [4061749, 3236400, 1988053, 1987896])]),
                  modif([])],
                 [head([termconcept(directory, [3214387])]),
                  modif([])]]]]]
```

e_2: "delete a text file"

```
Frame2 = [[[action,
                [[head([termconcept(delete,
                          [633190, 398344, 70485])]),
                  modif([])]]],
               [object,
                [[head(
                     [termconcept(file,
                         [4061749, 3236400, 1988053, 1987896])]),
                  modif(
                     [termconcept(text,
                         [3210873, 3201596])])]]]]]
```

e_3: "remove text from a file"

```
Frame3 = [[[action,
                [[head([termconcept(remove,
                          [983048, 951139, 951020, 179903, 70485])]),
                  modif([])]]],
               [object,
                [[head(
                     [termconcept(text, [3210873, 3201596])])],
                  modif([])]]],
               [source,
                [[head(
                     [termconcept(file,
                         [4061749, 3236400, 1988053, 1987896])]),
                  modif([])]]]]
```

Figura 5.12: Análisis de expresiones.

Ya que las cabezas tienen en este caso un solo término-concepto:

$$simHead(h_{f_1 action1}, h_{f_2 action1}) = simTermConcept(tc_{remove[..]}, tc_{delete[..]})$$
$$simTermConcept(tc_{remove[..]}, tc_{delete[..]}) = w_{remove} \cdot w_{delete} \cdot \theta(tc_{remove[..]}, tc_{delete[..]})$$
$$simTermConcept(tc_{remove[..]}, tc_{delete[..]}) = 4.8074 \cdot 8.7142 \cdot 1 = 41.8926$$

En donde los pesos de los término-conceptos se han calculado mediante la fórmula del logaritmo a partir del análisis de las descripciones

$$w_i = log_2(n/df_i)$$

y el valor de $\theta\,(tc_{remove[..]}, tc_{delete[..]})$ es 1, ya que:

$$conc(tc_{remove[..]}) \cap conc(tc_{delete[..]}) = \{70485\} \neq \varnothing$$

Con esto se obtiene como valor de similitud para los slots correspondientes a las acciones:

$$simSlot(s_{f_1 action}, s_{f_2 action}) = 41.8926$$

Para los slots correspondientes a los objetos, ya que en este caso el slot *object* del frame f_1 tiene dos fillers:

$$simSlot(sl_{f_1 object}, sl_{f_2 object}) = simFiller(fl_{f_1 object1}, fl_{f_2 object1}) + simFiller(fl_{f_1 object2}, fl_{f_2 object1})$$

La similitud entre los fillers, ya que las cabezas del de f_1 no tienen modificadores:

$$simFiller(fl_{f_1 object1}, fl_{f_2 object1}) = simHead(h_{f_1 object1}, h_{f_2 object1})$$
$$simFiller(fl_{f_1 object2}, fl_{f_2 object1}) = simHead(h_{f_1 object2}, h_{f_2 object1})$$

con la parte *head* de los slots:

$$h_{f_1 object1} = head([termconcept(file,[4061749, 3236400, 1988053, 1987896])])$$
$$h_{f_1 object2} = head([termconcept(directory,[3214387])])$$
$$h_{f_2 object1} = head([termconcept(file,[4061749, 3236400, 1988053, 1987896])])$$

La similitud entre las cabezas queda:

$$simHead(h_{f_1 object1}, h_{f_2 object1}) = simTermConcept(tc_{file[..]}, tc_{file[..]})$$
$$simHead(h_{f_1 object2}, h_{f_2 object1}) = simTermConcept(tc_{directory[..]}, tc_{file[..]})$$

Las similitud entre las parejas de término-conceptos es:

$$simTermConcept(tc_{file[..]}, tc_{file[..]}) = w_{file} \cdot w_{file} \cdot \theta(tc_{file[..]}, tc_{file[..]})$$
$$simTermConcept(tc_{file[..]}, tc_{file[..]}) = 2.1444 \cdot 2.1444 \cdot 1 = 4.5985$$

$$simTermConcept(tc_{directory[..]}, tc_{file[..]}) = w_{directory} \cdot w_{file} \cdot \theta(tc_{directory[..]}, tc_{file[..]})$$
$$simTermConcept(tc_{directory[..]}, tc_{file[..]}) = 5.3923 \cdot 2.1444 \cdot 0 = 0$$

En donde se han incluido los pesos de los términos calculados utilizando la fórmula del logaritmo, y los valores de la función θ para las parejas de términos-concepto, que son 1 y 0 respectivamente, ya que:

$$conc(tc_{file[..]}) \cap conc(tc_{file[..]}) = \{4061749, 3236400, 1988053, 1987896\} \neq \varnothing$$
$$conc(tc_{directory[..]}) \cap conc(tc_{file[..]}) = \varnothing$$

Con esto se obtiene como valor de similitud para los slots correspondientes a los objetos:

$$simSlot(sl_{f_1 object}, sl_{f_2 object}) = 4.5985$$

De esta forma, la similitud entre los frames f_1 y f_2, y con ello la de las expresiones e_1 y e_2, queda finalmente:

$$simFrame(f_1, f_2) = 46.4911$$

b) Cálculo de la similitud entre las expresiones e_1 y e_3.

La similitud entre los frames f_1 y f_3 que representan las expresiones e_1 y e_3 es:

$$simFrame(f_1, f_3) = simSlot(sl_{f_1 action}, sl_{f_3 action}) + simSlot(sl_{f_1 object}, sl_{f_3 object})$$

ya que aunque f_3 tiene slot *source*, f_1 no lo tiene.

Para la similitud de los slots correspondientes a las acciones, ya que cada slot tiene un único filler, y ninguno de los fillers tiene modificadores:

$$simSlot(sl_{f_1 action}, sl_{f_3 action}) = simFiller(fl_{f_1 action1}, fl_{f_3 action1})$$
$$simFiller(fl_{f_1 action1}, fl_{f_3 action1}) = simHead(h_{f_1 action1}, h_{f_3 action1})$$

Con la parte *head* de los fillers:

$$h_{s_{f_1 action}} = h_{s_{f_3 action}} = head([termconcept(remove, [983048, 951139, 951020, 179903, 70485])])$$

Ya que las cabezas tienen un solo término-concepto y además es el mismo:

$$simHead(h_{f_1 action1}, h_{f_3 action1}) = simTermConcept(tc_{remove[..]}, tc_{remove[..]})$$
$$simTermConcept(tc_{remove[..]}, tc_{remove[..]}) = w_{remove} \cdot w_{remove} \cdot \theta(tc_{remove[..]}, tc_{remove[..]})$$
$$simTermConcept(tc_{remove[..]}, tc_{remove[..]}) = 4.8074 \cdot 4.8074 \cdot 1 = 23.1111$$

En donde se han incluido los pesos de los términos calculados utilizando la fórmula del logaritmo, y el valor de la función θ es 1 ya que los conjuntos de conceptos asociados a cada término concepto son el mismo y su intersección es no nula.

De esta forma se obtiene como valor de similitud para los slots correspondientes a las acciones:

$$simSlot(s_{f_1 action}, s_{f_3 action}) = 23.1111$$

Para los slots correspondientes a los objetos, ya que el slot *object* del frame f_1 tiene dos fillers y el de f_3 tiene solo uno:

$$simSlot(sl_{f_1 object}, sl_{f_3 object}) = simFiller(fl_{f_1 object1}, fl_{f_3 object1}) + simFiller(fl_{f_1 object2}, fl_{f_3 object1})$$

La similitud entre los fillers, ya que ninguno tiene modificadores:

$$simFiller(fl_{f_1 object1}, fl_{f_3 object1}) = simHead(h_{f_1 object1}, h_{f_3 object1})$$
$$simFiller(fl_{f_1 object2}, fl_{f_3 object1}) = simHead(h_{f_1 object2}, h_{f_3 object1})$$

con la parte *head* de los slots:

$$h_{f_1 object1} = head([termconcept(file,[4061749,3236400,1988053,1987896])])$$
$$h_{f_1 object2} = head([termconcept(directory,[3214387])])$$
$$h_{f_3 object1} = head([termconcept(text,[3210873,3201596])])$$

La similitud entre las cabezas queda:

$$simHead(h_{f_1 object1}, h_{f_3 object1}) = simTermConcept(tc_{file[..]}, tc_{text[..]})$$
$$simHead(h_{f_1 object2}, h_{f_3 object1}) = simTermConcept(tc_{directory[..]}, tc_{text[..]})$$

Las similitudes entre los término-conceptos son:

$$simTermConcept(tc_{file[..]}, tc_{text[..]}) = w_{file} \cdot w_{text} \cdot \theta(tc_{file[..]}, tc_{text[..]})$$
$$simTermConcept(tc_{file[..]}, tc_{text[..]}) = 2.1444 \cdot 5.2548 \cdot 0 = 0$$

$$simTermConcept(tc_{directory[..]}, tc_{text[..]}) = w_{directory} \cdot w_{text} \cdot \theta(tc_{directory[..]}, tc_{text[..]})$$
$$simTermConcept(tc_{directory[..]}, tc_{text[..]}) = 5.3923 \cdot 5.2548 \cdot 0 = 0$$

En donde, como en los cálculos anteriores, se han incluido los pesos de los términos calculados utilizando la fórmula del logaritmo, y los valores de la función θ para las parejas de términos-concepto, que son 0 en ambos casos, ya que:

$$conc(tc_{file[..]}) \cap conc(tc_{text[..]}) = conc(tc_{directory[..]}) \cap conc(tc_{text[..]}) = \varnothing$$

Con esto se obtiene 0 como valor de similitud para los slots correspondientes a los objetos:

$$simSlot(sl_{f_1 object}, sl_{f_3 object}) = 0$$

De esta forma, la similitud entre los frames f_1 y f_3, y con ello la de las expresiones e_1 y e_3, queda finalmente:

$$simFrame(f_1, f_3) = 23.1111$$

5.8 Experimentos centrados en la efectividad

En este punto describimos una serie de experimentos orientados al estudio de la efectividad del sistema Ares. Para ello hemos considerado dos cuestiones principalmente. En primer lugar, nos interesa comparar la efectividad del sistema con respecto a una aproximación basada en el modelo vectorial, uso de pesos de términos, extracción de raíces y lista de parada. En segundo lugar, los experimentos se encuentran también orientados al estudio del comportamiento de los elementos con información lingüística que incluye Ares: el léxico y la gramática de unificación.

Para los dos fines anteriormente señalados, hemos desarrollado dos sistemas adicionales, con cuyo comportamiento comparamos el de Ares. El primero se basa en el modelo del espacio vectorial y el segundo utiliza parte de la información lingüística incluida en Ares: la existente en el léxico.

Nuestros experimentos centrados en la efectividad, se basan en medidas de los índices de *recall* y *precision*. Para la medida de ambos parámetros hemos decidido emplear un conjunto de consultas de usuarios basado en uno utilizado ya para el estudio de la efectividad en otros trabajos realizados sobre el dominio de las órdenes del S.O. UNIX [Maarek91, Frakes94].

Tratamos a continuación cómo se ha obtenido el conjunto de consultas utilizado en los experimentos, los sistemas que se han desarrollado con este fin, y los resultados experimentales obtenidos.

5.8.1 La colección experimental de documentos y consultas

En el planteamiento de experimentos centrados en la efectividad de sistemas de RI, se parte habitualmente de un conjunto de documentos y un conjunto de consultas con asignaciones previas de relevancias [Salton83, Fagan87]. Algunas colecciones de documentos y consultas, adoptadas como estándares de hecho, y que se utilizan con estos fines son CACM, CISI, CRAN, MED e INSPEC. La colección CACM contiene 3.204 documentos correspondientes a artículos de la publicación "Communications of the Association for Computing Machinery" entre los años 1958 y 1979, y 52 consultas orientadas a este dominio. CISI contiene documentos y consultas relacionados con Ciencias de la Información. CRAN, documentos relativos a Aerodinámica e Ingeniería Aeronáutica. MED, relacionados con Medicina, e INSPEC con Electrónica [Fagan87, Lewis92].

La especificidad del dominio de Ares, implica que no resulte factible la utilización de las anteriores colecciones de documentos y consultas. En el desarrollo de Ares, la definición del dominio ha sido uno de los puntos de partida, por lo que la colección de documentos sobre la que realizar los experimentos centrados en la efectividad se ha encontrado determinada desde el principio. Por otra parte, resulta necesario el desarrollo de un conjunto de consultas de usuarios, con criterios de relevancia previamente asignados a los documentos de la colección de este dominio específico.

Para los experimentos que presentamos en este punto hemos utilizado el conjunto de consultas con relevancias asignadas desarrollado para el estudio del sistema Guru [Maarek91], tratado en un capítulo anterior. Este sistema está orientado al mismo dominio que Ares: el manual del S.O. UNIX. En otros trabajos incluidos en la bibliografía [Frakes94] se utiliza a su vez este conjunto de consultas como base para la medida experimental de valores centrados en la efectividad de otros sistemas, como Proteus. Disponer de este conjunto de consultas, específico de nuestro dominio, nos ha facilitado el desarrollo de los experimentos que presentamos, además de proporcionar una objetividad adicional frente a un posible conjunto de consultas desarrollado íntegramente por nosotros.

El conjunto de consultas de Maarek se encuentra orientado a AIX (la versión de UNIX de IBM), mientras que en Ares hemos utilizado la sección 1 del manual de SunOS. Ambos sistemas operativos tienen un conjunto de órdenes comunes importante pero tienen también ciertas diferencias. Esto ha hecho necesaria una adaptación a nuestro dominio concreto, al igual que se ha realizado en el trabajo de Frakes, anteriormente referenciado. En concreto, se han eliminado 8 de las 30 consultas incluidas en el conjunto original, por referenciar órdenes específicas de AIX que no se encuentran en SunOS (e.g. *chcursor*, *del*) o bien por referenciar órdenes que en SunOS se encuentran en la sección 8 del manual (órdenes de mantenimiento) y no en la sección 1 que es la que hemos considerado en el desarrollo de Ares (e.g. *chown*, *route*). En la figura 5.13. aparece el conjunto de consultas utilizado en los experimentos, acompañadas de las órdenes relevantes a cada una de ellas.

De esta forma, el conjunto de documentos (las 432 descripciones cortas de las órdenes de la sección 1 del manual de SunOS) y el conjunto de consultas con órdenes relevantes asociadas de la figura 5.13, constituye un conjunto de prueba análogo a los anteriormente mencionados CACM, CISI, CRAN, MED e INSPEC, pero específico de nuestro dominio.

Una limitación que tiene este conjunto de prueba -heredada del conjunto de consultas de Maarek- respecto a los estándares es el número de documentos relevantes asociados a cada consulta: oscila entre 1 y 4, mientras que en las colecciones estándares suele ser mayor. Esta limitación hace que determinadas operaciones orientadas a la medida de la efectividad, como la obtención de curvas precision-recall a partir de la extrapolación de puntos concretos, no resulten posibles, aunque sí resulta posible la obtención de valores medios de precision para un recall determinado.

En concreto, 18 de las 22 consultas tienen asignado sólo un documento relevante. Para estas consultas el recall sólo puede tomar valores binarios, esto es {0, 1}. Sólo para las dos 2 consultas que tienen 2 documentos relevantes asignados el recall puede tomar valores {0, 0.5, 1}, y sólo para las 2 consultas que tienen asignados 4 puede tomar valores {0 ,0.25, 0.5, 0.75, 1}. De esta forma los valores realmente significativos que pueden ser calculados para la precisión son los obtenidos para recall = 1.

Ref.	Consulta	Orden relevante
1.	compare two files	diff, cmp
2.	join columns of two different files into a single file	join
3.	delete a file	rm
4.	change the permission access to a file	chmod
5.	rename a directory	mv
6.	overwrite a file	mv
7.	put a file in another directory	mv
8.	get the name of the current directory	pwd
9.	create a new directory	mkdir
10.	become the root user	su
11.	get information about the type of a file	file
12.	distinguish between ascii and binary files	file
13.	change my password	passwd
14.	display when a file was last written to	ls
15.	set time and date of the clock	date
16.	copy a file from a remote host	rcp, ftp
17.	format C source code	cb
18.	stop a process	kill
19.	split files according to a pattern parameter	csplit
20.	send binary files via electronic mail	uuencode
21.	locate a regular expression in a file	awk, grep, egrep, fgrep
22.	look for a pattern in a file	awk, grep, egrep, fgrep

Figura 5.13: Colección de consultas.

En la figura 5.14 se presentan datos estadísticos de las colecciones de documentos y consultas CACM, CRAN Y MED [Fagan87], los correspondientes calculados para la colección utilizada en nuestros experimentos (referenciada como "Ares") y los de los utilizados en los estudios de los sistemas FERRET [Mauldin91] y UCF [Wendlandt91]. En "términos" se incluye el número de términos obtenido después de la aplicación de extracción de raíces a los corpus de palabras correspondientes.

Por lo que respecta a la colección de documentos, el número de ellos utilizado se encuentra dentro del mismo orden de magnitud, destacando el diferente valor del número de términos medio por documento. Este valor hace patente la brevedad de las descripciones de las órdenes de UNIX en comparación con los documentos incluidos en otras colecciones, así como la mayor longitud de los documentos del dominio de FERRET, en los que, como ya se ha comentado, una aproximación basada en text skimming resulta adecuada.

En cuanto al conjunto de consultas, el número de ellas también se encuentra dentro de los mismos órdenes de magnitud, y su longitud media es también menor que las de otras colecciones, pero la diferencia es menor que en el caso de las descripciones.

Colección de Documentos	CACM	CRAN	MED	ARES	FERRET	UCF
Número de documentos	3204	1398	1033	432	532	160
Número de términos	4522	3763	6927	604	-	-
Térms. medios documento	20.22	53.13	51.60	6.39	300	-
Colección de Consultas	**CACM**	**CRAN**	**MED**	**ARES**	**FERRET**	**UCF**
Número de consultas	52	225	30	22	22	21
Número de términos	324	585	241	72	-	-
Térms. medios consulta	10.67	9.17	10.10	5.59	-	-

Figura 5.14: Estadísticas de colecciones de documentos y consultas.

5.8.2 Métodos de recuperación utilizados en los experimentos

Para el estudio del comportamiento centrado en la efectividad del sistema Ares, hemos desarrollado dos sistemas adicionales adaptados a nuestro dominio. Con estos sistemas hemos realizado las mismas medidas basadas en la utilización del conjunto de consultas y documentos descrito anteriormente, para su comparación con los resultados obtenidos con Ares.

El primer sistema implementa una aproximación basada en el modelo del espacio vectorial, uso de pesos de términos, extracción de raíces y lista de parada. Este sistema representa la aproximación seguida en Argos, con modificaciones para adaptarlo al caso de las descripciones cortas. La efectividad proporcionada por este sistema nos sirve como referencia para la comprobación de las posibles mejoras conseguidas con Ares.

El segundo sistema se encuentra orientado al estudio del comportamiento de cada uno de los dos componentes fundamentales de Ares que incluyen la información lingüística que utiliza: el léxico y la gramática de unificación. Este segundo sistema se basa, al igual que el primero, en la aproximación del modelo del espacio vectorial, uso de pesos de términos, extracción de raíces y lista de parada y, además, utiliza la información léxica obtenida de WordNet relacionada con la sinonimia entre términos. Ambos sistemas los referiremos por los nombres EV y EV-S, respectivamente, en los párrafos siguientes. A continuación describimos sus principales aspectos.

El sistema EV

El sistema EV se basa en el modelo del espacio vectorial, uso de pesos de términos, extracción de raíces y lista de parada. El sistema EV se ha desarrollado en Prolog y accede a las mismas descripciones y consultas que Ares. Para el desarrollo del sistema EV se han reutilizado determinados elementos de Ares e incluido en él

adaptaciones al dominio de las descripciones cortas con respecto a los elementos utilizados en Argos.

Para la implementación del algoritmo de extracción de raíces hemos utilizado la componente morfológica de la gramática de unificación de Ares. El algoritmo de extracción de raíces se basa en el predicado *morphie*, de acceso al analizador morfológico. El siguiente es un ejemplo de la extracción de raíces a través de *morphie*.

```
?- morphie(change,Stem,_).
    Stem = change
?- morphie(changes,Stem,_).
    Stem = change
?- morphie(changing,Stem,_).
    Stem = change
```

El algoritmo de extracción de raíces implementado mediante el analizador morfológico proporciona una mayor precisión que los convencionales, como el de Porter [Frakes92], utilizado en Argos. El algoritmo de extracción de raíces utiliza como base el léxico de Ares y no se comenten eliminaciones de sufijos incorrectas al comprobar la existencia del término correspondiente a la raíz en el léxico (i.e. se reducen los errores de infrarradicación y sobrerradicación).

La lista de palabras vacías utilizada en EV también se ha adaptado al dominio. Listas de palabras vacías de propósito general con un gran número de términos, como la de 429 términos [Fox92] utilizada en Argos, no resulta adecuada para el conjunto de descripciones cortas. Términos que presentan frecuencias de aparición muy alta en otros dominios y se incluyen en estas listas de palabras vacías, sí son adecuados como índices en el nuestro y no deben excluirse (e.g. "clear", "group" y "working"). De esta forma, hemos utilizado como lista de palabras vacías, más adecuada a nuestro dominio y a la condición de brevedad a las descripciones, una lista de palabras vacías con un reducido número de elementos. En concreto, hemos utilizado la empleada en el sistema comercial ORBIT, formada por ocho palabras [Fox92]:

a	an	and	by
from	of	the	with

El cálculo de la similitud se basa en el modelo del espacio vectorial y en el cálculo de pesos de términos. Como función de peso de los términos hemos utilizado la basada en el logaritmo del inverso de la frecuencia de aparición de los términos en documentos [Salton89] también utilizada en Argos:

$$w_i = log_2 (n / df_i)$$

En donde n es el número de documentos de la colección y df_i el número de documentos en que aparece el término i. El cálculo de estos pesos se basa en predicados que ya implementan esta funcionalidad en Ares para el cálculo de la similitud de las asociaciones término-concepto en las que se utilizan estos pesos.

Como en el módulo de RI de Argos, cada documento queda representado, o indexado, por un vector de dimensión m, con los pesos asignados a cada uno de términos de indexación tras la utilización de la lista de parada y la extracción de raíces. El vector que representa el documento j es:

$$(wd_{j1}, wd_{j2}, ..., wd_{jm})$$

El procesamiento de las consultas se realiza de una forma análoga. quedando representada por un vector de dimensión m, con los pesos asignados a cada uno de términos de indexación. El vector que representa la consulta k es:

$$(wq_{k1}, wq_{k2}, ..., wq_{km})$$

Para el cálculo de la similitud entre un documento d_j y una consulta q_k se utiliza la fórmula del coseno que mide el grado de ortoganilidad de ambos vectores:

$$sim(d_j, q_k) = \frac{\sum_{i=1}^{m} wd_{ji} \cdot wq_{ki}}{\sqrt{\sum_{i=1}^{m} wd_{ji}^2 \cdot \sum_{i=1}^{m} wq_{ki}^2}}$$

En el apéndice final se incluye el detalle de la implementación del sistema EV.

El sistema EV-S

El sistema EV-S se basa en el sistema EV -modelo del espacio vectorial, extracción de raices y lista de parada- y además utiliza la información existente en el léxico de Ares relativa a la sinonimia. El sistema EV-S se ha desarrollado en Prolog y se basa en gran medida en EV. La diferencia con EV es la utilización de la información relativa a sinonimia de los términos que se encuentra en el léxico de Ares. La información utilizada en EV-S se corresponde esencialmente con las asociaciones término-término-concepto que se presentaron en la figura 5.7.

El sistema EV-S realiza el mismo análisis de documentos, de las consultas y cálculo de la similitud que EV, con la salvedad de que cuando aparecen términos diferentes, pero que tienen asociados un concepto común (esto es, son sinónimos), también contribuyen en el valor de la similitud mediante el producto de sus pesos, como en la expresión anterior para términos que son iguales, esto es:

$$sim(d_j, q_k) = \frac{\sum_{i=1}^{m} \sum_{l=1}^{m} wd_{ji} \cdot wq_{kl} \cdot sin(i,l)}{\sqrt{\sum_{i=1}^{m} wd_{ji}^2 \cdot \sum_{i=1}^{m} wq_{ki}^2}}$$

En donde la función $sin(i,l)$ es tal que es la unidad cuando el término i y el término l son el mismo o son sinónimos, y nula en caso contrario. El comportamiento del sistema EV-S nos permite estudiar, mediante su comparación con el de EV, la

forma en que influye la información léxica obtenida de WordNet relativa a la sinonimia de términos en el proceso de recuperación. La comparación del comportamiento del sistema EV-S con Ares, nos permitirá caracterizar esencialmente, la mejora en la precisión que introduce el componente de Ares basado en la gramática de unificación.

Los sistemas MV y MV-S se han implementado como módulos adicionales sobre la estructura de Ares representada en la figura 5.6. Los resultados de los valores de la similitud de los documentos para cada consulta se han almacenado en el archivo correspondiente al elemento de datos *descriprelevantes.pro* de la figura 5.6. Los datos se han almacenado en forma de hechos Prolog, constituyendo de esta forma una base de datos con valores relativos a la similitud de consultas y documentos que facilita su manipulación y la realización de cálculos orientados a la interpretación de los resultados obtenidos.

5.8.3 Resultados experimentales e interpretación

Utilizando la colección de documentos y consultas los tres sistemas proporcionan los valores de similitud correspondientes. En los tres sistemas, los valores de similitud se utilizan como base para el ordenamiento de los documentos por relevancia asignada por el sistema (ranking). En el caso de descripciones con valores de similitud iguales para una consulta, se presentan por orden alfabético en los tres sistemas. Consideramos documentos recuperados por los sistemas aquellos a los que se les asigna una similitud no nula.

En la figura 5.15 se incluyen las cinco primeras consultas del conjunto de prueba con las descripciones de sus órdenes relevantes asociadas En las tablas de las figuras 5.16, 5.17 y 5.18 se presentan los ordenamientos por relevancia de las órdenes proporcionados por los tres sistemas. En estas tablas se encuentran destacados en negrita y mayúsculas los nombres de las órdenes relevantes. El conjunto de documentos recuperados y las similitudes calculadas por los sistemas para la totalidad de las consultas puede encontrarse en el apéndice final.

En estas tablas relativas a las cinco primeras consultas se pueden advertir algunas de las características generales de cada uno de los sistemas. Estas características se observan también para el conjunto total de consultas. En primer lugar, el número de documentos recuperados por MV-S es siempre mayor que el de MV, y este a su vez suele ser mayor que el de Ares. Por otra parte, el comportamiento de los sistemas es parecido en algunas consultas, como la 4, en la que los tres sistemas proprocionan el documento relevante en primera posición en el ranking, o la 2, en la que ninguno identifica acertadamente el documento relevante. Sin embargo, en otras consultas el comportamiento de los sistemas varía entre sí importantemente.

Por ejemplo, en la consulta 3, el sistema que proporciona mejores resultados es MV-S, seguido de Ares, y después MV. En esta consulta, para un valor de *recall* = 1, los valores obtenidos por Ares, MV y MV-S para la *precision* son de 0.500, 0.333, y 1.000, respectivamente. En la consulta 5 sin embargo, Ares y MV proporcionan en primer lugar el documento relevante, mientras que MV-S en segundo. En esta

Ref.	Consulta	Orden relevante	Descripción
1.	compare two files	diff	display line-by-line differences between pairs of text files
		cmp	perform a byte-by-byte comparison of two files
2.	join columns of two different files into a single file	join	relational database operator
3.	delete a file	rm	remove (unlink) files or directories
4.	change the permission access to a file	chmod	change the permissions mode of a file
5.	rename a directory	mv	move or rename files

Figura 5.15: Muestra de consultas y descripciones de órdenes relevantes.

consulta, para *recall* = 1, los valores obtenidos por Ares, MV y MV-S para la *precision* son de 1.000, 1.000 y 0.500, respectivamente.

A partir de las ordenaciones por relevancia proporcionadas por los sistemas para el conjunto completo de consultas, se obtienen los datos de la tabla que se presenta en la figura 5.19. En ella NRels representa el número de documentos relevantes a la consulta asignados en el conjunto de prueba. NRets representa el número de documentos recuperados por los sistemas, esto es, con similitud no nula. NRetRels, el número de documentos recuperados que son realmente relevantes. PosRetRels, las posiciones en el ranking en que han aparecido los documentos realmente relevantes.

En esta tabla se observa sobre el conjunto total de consultas, la diferencia existente en el número de documentos recuperados con similitud no nula entre los sistemas (NRets), esto se debe a que, por la propia definición de los sistemas, MV-S tiene que proporcionar siempre un superconjunto de la unión de los proporcionados por MV y Ares, es decir:

$$Rets_{Ares} \cup Rets_{MV} \subseteq Rets_{MVS}$$

y siempre se cumple:

$$NRets_{Ares} \leq NRets_{MVS}$$
$$NRets_{MV} \leq NRets_{MVS}$$

De acuerdo con esto, sobre los datos experimentales se obtiene que el número de documentos relevantes recuperados, NRetRels, es mayor para MV-S, seguido por MV y Ares en último lugar.

Ref.	Documentos Relevantes
1.	sccs-sccsdiff, sccsdiff, diremp, acctcom, admin, ar, bar, checknr, compress, cp, crontab, crypt, csplit, ctags, dd, dos2unix, du, egrep, fgrep, grep, indent, install, make, mv, pack, pcat, pr, rm, rmdir, sccs-admin, sccs-val, split, tar, uncompress, unix, unpack, uudecode, uuencode, uusend, val, vswap, zcat
2.	paste, adjacentscreens, ld, ld.so, sccs-sccsdiff, sccsdiff, chgrp, chmod, cpio, delta, df, error, get, head, paxcpio, sact, sccs-comb, sccs-get, sccs-show, screendump, size, strings, tail, touch
3.	crontab, RM, rmdir, colrm, ipcrm, lprm, rmdel, sccs-rmdel, strip, unifdef, uniq, unwhiteout, acctcom, admin, ar, bar, checknr, compress, cp, crypt, csplit, ctags, dd, dos2unix, du, egrep, fgrep, grep, indent, install, make, mv, pack, pcat, pr, sccs-admin, sccs-val, split, tar, uncompress, unix, unpack, uudecode, uuencode, uusend, val, vswap, zcat
4.	CHMOD, lockscreen, touch, mv, coloredit, env, stty, dd, dos2unix, old-sun3cvt, ranlib, unix, vswap, cd, cdc, chfn, chgrp, chkey, chsh, passwd, sccs-cdc, sv_acquire, sv_release, yppasswd, delta, sccs-comb, screendump, strings, sum
5.	MV, cd, coloredit, env, stty, dd, dos2unix, old-sun3cvt, ranlib, unix, vswap, cdc, chfn, chgrp, chkey, chmod, chsh, passwd, sccs-cdc, sv_acquire, sv_release, yppasswd, diremp, mkdir, rm, rmdir

Figura 5.16: Sistema Ares.Ordenación por relevancia de documentos

Ref.	Documentos Relevantes
1.	sccs-sccsdiff, sccsdiff, diremp, comm, CMP, sdiff, rastrepl, cp, rm, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, chgrp, compress, uncompress, zcat, admin, sccs-admin, mv, sum, tftp, chmod, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, DIFF, touch, cut, head, old-filemerge, checknr, dos2unix, unix, ln, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, paste, error, du, pr
2.	paste, sccs-sccsdiff, sccsdiff, pr, colrm, cmp, comm, sdiff, rastrepl, cp, rm, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, chgrp, compress, uncompress, zcat, admin, sccs-admin, mv, sum, tftp, chmod, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, diff, touch, cut, head, old-filemerge, checknr, dos2unix, unix, ln, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, error, du
3.	keylogout, cp, RM, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, chgrp, compress, uncompress, zcat, sccs-sccsdiff, sccsdiff, admin, sccs-admin, mv, sum, tftp, chmod, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, diff, touch, cut, head, old-filemerge, checknr, dos2unix, unix, ln, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, paste, error, du, cmp, sdiff, pr
4.	CHMOD, touch, lockscreen, chgrp, cd, chkey, cdc, sccs-cdc, chfn, chsh, passwd, yppasswd, sv_acquire, sv_release, cp, rm, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, compress, uncompress, zcat, sccs-sccsdiff, sccsdiff, admin, sccs-admin, mv, sum, tftp, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, diff, cut, head, old-filemerge, checknr, dos2unix, unix, ln, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, paste, error, du, cmp, sdiff, pr
5.	MV, rm, rmdir, mkdir, diremp, ls, organizer, cd, pwd, du, whois

Figura 5.17: Sistema MV. Ordenación por relevancia de documentos

Ref.	Documentos Relevantes
1.	sccs-sccsdiff, sccsdiff, diremp, DIFF, comm, CMP, sdiff, rastrepl, cp, rm, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, chgrp, compress, uncompress, zcat, admin, sccs-admin, mv, sum, tftp, chmod, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, touch, cut, head, old-filemerge, checknr, dos2unix, unix, ln, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, paste, error, du, pr
2.	ld, ld.so, paste, ypmatch, cu, tip, find, line, talk, write, ln, diff, sccs-sccsdiff, sccsdiff, adjacentscreens, toolplaces, whatis, pr, colrm, yacc, cmp, comm, sdiff, rastrepl, cp, rm, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, chgrp, compress, uncompress, zcat, admin, sccs-admin, mv, sum, tftp, chmod, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, touch, cut, head, old-filemerge, checknr, dos2unix, unix, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, error, du
3.	RM, rmdir, rmdel, sccs-rmdel, keylogout, cancel, lp, overview, lprm, cut, unwhiteout, strip, crontab, uniq, atrm, colrm, deroff, unifdef, ipcrm, cp, mkstr, rcp, install, pack, pcat, unpack, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, chgrp, compress, uncompress, zcat, sccs-sccsdiff, sccsdiff, admin, sccs-admin, mv, sum, tftp, chmod, dd, vswap, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, diff, touch, head, old-filemerge, checknr, dos2unix, unix, ln, pg, make, objdump, diffmk, strings, df, uudecode, uuencode, ctags, csplit, paste, error, du, cmp, sdiff, pr
4.	CHMOD, touch, mesg, mv, pax, lockscreen, stty, dd, vswap, chgrp, cd, env, coloredit, dos2unix, unix, ranlib, chkey, cdc, sccs-cdc, chfn, chsh, passwd, yppasswd, sv_acquire, sv_release, old-sun3cvt, rasfilter, cp, rm, rmdir, mkstr, rcp, install, pack, pcat, unpack, rmdel, sccs-rmdel, sccs-val, val, ftp, delta, sccs-comb, uusend, size, crypt, organizer, get, sccs-get, uupick, uuto, tail, rdist, what, cpio, paxcpio, prt, sccs-prt, split, compress, uncompress, zcat, sccs-sccsdiff, sccsdiff, admin, sccs-admin, sum, tftp, egrep, fgrep, grep, more, page, sccs-unget, unget, old-ccat, old-compact, old-uncompact, find, acctcom, sact, sccs-show, indent, bar, tar, ar, file, screendump, screenload, cluster, diff, cut, head, old-filemerge, checknr, ln, pg, make, objdump, diffmk, strings, strip, df, uudecode, uuencode, ctags, csplit, crontab, paste, error, du, cmp, sdiff, pr
5.	cd, MV, pax, stty, env, dd, vswap, coloredit, rm, rmdir, chgrp, mkdir, ranlib, chkey, chmod, cdc, sccs-cdc, chfn, chsh, passwd, dos2unix, unix, diremp, yppasswd, ls, sv_acquire, sv_release, organizer, old-sun3cvt, rasfilter, pwd, du, whois

Figura 5.18: Sistema MV-S. Ordenación por relevancia de documentos

	NRls	NRets			NRetRels			PosRetRels		
Ref.		Ares	MV	MVS	Ares	MV	MVS	Ares	MV	MVS
1.	2	42	98	98	0	2	2	-	5,74	4,6
2.	1	24	98	110	0	0	0	-	-	-
3.	1	48	96	107	1	1	1	2	3	1
4.	1	29	106	114	1	1	1	1	1	1
5.	1	26	11	33	1	1	1	1	1	2
6.	1	55	95	124	1	1	1	40	41	73
7.	1	74	129	138	1	1	1	48	78	89
8.	1	36	31	81	1	1	1	1	4	4
9.	1	20	23	30	1	1	1	1	4	1
10.	1	6	16	30	0	1	1	-	1	16
11.	1	44	117	138	0	1	1	-	17	43
12.	1	0	97	112	0	1	1	-	70	87
13.	1	24	12	29	1	1	1	21	3	3
14.	1	0	142	148	0	0	0	-	-	-
15.	1	11	16	30	1	1	1	1	2	3
16.	2	44	113	113	0	2	2	-	1,40	1,40
17.	1	9	26	34	0	1	1	-	19	27
18.	1	2	8	16	1	1	1	1	2	1
19.	1	0	99	99	0	1	1	-	3	5
20.	1	46	112	120	1	1	1	2	8	22
21.	4	18	125	125	3	3	3	5,6,7	1,2,3	1,2,3
22.	4	16	123	128	3	4	4	2,3,4	1,77,78,79	1,2,3,8

Figura 5.19: Estadísticas de documentos relevantes y recuperados

Sin embargo, la información más significativa para el estudio de la efectividad es la que se encuentra en las columnas correspondientes a PosRetRels, en la que se indica en qué posición se presentan los documentos en el ranking. Aquí puede observarse que existen consultas en las que los sistemas, especialmente MV y MV-S, recuperan los documentos relevantes, pero quedando en posiciones muy lejanas del ranking (e.g. consultas 6,7,11,12) y esto a efectos de la efectividad proporciona incrementos pequeños (i.e. un documento relevante presentado en la posición 87 del ranking no es tenido en cuenta por el usuario).

Finalmente, en la tabla de la figura 5.20, se incluyen los valores definitivos de precision para *recall* = 1.000 para los tres sistemas. Estos datos nos muestran que los valores medios para los sistemas son:

Ares: $\text{Precision}_{\text{Recall=1.000}} = 32.24\%$
MV: $\text{Precision}_{\text{Recall=1.000}} = 26.54\%$
MV-S: $\text{Precision}_{\text{Recall=1.000}} = 27.90\%$

Estos datos suponen un incremento absoluto de un 5.74%, y relativo del 21.5% del valor de precision para *recall* = 1 de Ares respecto a MV. Por otra parte, los resultados entre MV y MV-S varían de una consulta a otra y en media, de una

Ref.	$Precision_{R=1.000}$		
	Ares	MV	MVS
1.	0.0000	0.0135	0.1666
2.	0.0000	0.0000	0.0000
3.	0.5000	0.3333	1.0000
4.	1.0000	1.0000	1.0000
5.	1.0000	1.0000	0.5000
6.	0.0250	0.0244	0.0137
7.	0.0208	0.0128	0.0112
8.	1.0000	0.2500	0.2500
9.	1.0000	0.2500	1.0000
10.	0.0000	1.0000	0.0625
11.	0.0000	0.0588	0.0233
12.	0.0000	0.0143	0.0115
13.	0.0476	0.3333	0.3333
14.	0.0000	0.0000	0.0000
15.	1.0000	0.5000	0.3333
16.	0.0000	0.0250	0.0250
17.	0.0000	0.0526	0.0370
18.	1.0000	0.5000	1.0000
19.	0.0000	0.3333	0.2000
20.	0.5000	0.1250	0.0455
21.	0.0000	0.0000	0.0000
22.	0.0000	0.0127	0.1250
V. M.	0.3224	0.2654	0.2790

Figura 5.18: Valores de $Precision_{Recall=1.000}$

forma menor que para Ares, suponen un incremento absoluto del 1.36% y un incremento relativo del 5.1%.

Estos resultados suponen una mejora considerable en la efectividad respecto a la aproximación basada en el modelo del espacio vectorial, uso de pesos de términos, extracción de raíces y lista de parada en nuestros experimentos.

La información relativa a sinonimia, de propósito general incluida en WordNet, exclusivamente, en la forma en que se ha realizado en MV-S, hace que para consultas determinadas se consigan mejoras en la efectividad, pero en otras, esta información añade ruido, en detrimento de la precision, de acuerdo con los resultados obtenidos en [Voorhees93]. Sin embargo, para nuestro dominio y el conjunto de consultas considerado, se consigue, en media, una mejora de la efectividad para esta última aproximación, aunque reducida.

La utilización de la información lingüística específica del dominio incluida en la gramática de unificación en Ares proporciona un mecanismo para la mejora de la precision que resulta decisiva para un incremento en la efectividad notablemente mayor respecto a un sistema como MV-S basado sólo en la información léxica (pasar del 5.1% al 21.5% relativos en los experimentos).

5.9 Resumen y conclusiones

En este capítulo se ha propuesto un modelo de aplicación de las técnicas de PLN en la RI orientado a la documentación existente en bibliotecas de componentes software. Se ha presentado el sistema Ares que materializa este modelo. El sistema se centra en los problemas que plantea la brevedad de las descripciones cortas existentes en las bibliotecas de componentes. Ares se ha desarrollado para las descripciones cortas de las órdenes de UNIX de la sección 1 del manual del sistema operativo. Este conjunto de descripciones está formado por 479 elementos y es representativo del caso general de las descripciones cortas.

En Ares se sigue una aproximación basada en la semántica. En el desarrollo del sistema se han marcado como objetivos fundamentales la mejora de la efectividad del proceso de recuperación y la disminución del coste de desarrollo del sistema. Además, la aproximación seguida en el sistema facilita su adaptación a otras bibliotecas de componentes diferentes de las órdenes de UNIX, como la que se ha realizado para el entorno de Smalltalk VisualWorks.

Para reducir el esfuerzo de desarrollo, se ha construido automáticamente el léxico del sistema utilizado la información lingüística de propósito general existente en una base de datos léxica, WordNet. Además se ha desarrollado el analizador-traductor mediante una gramática de unificación, en la que se incluye, básicamente, la información lingüística específica del dominio.

En la definición del lenguaje de representación de las descripciones y las consultas se ha buscado su adecuación a restricciones específicas del dominio del lenguaje procesado y de los objetos descritos: las ordenes de UNIX. La representación semántica de las expresiones se basa en los roles de las gramáticas de casos y los significados de los términos originalmente definidos en WordNet. La definición de la gramática se ha realizado de forma incremental hasta conseguir el procesamiento correcto del 64.8% de las descripciones. En el desarrollo de la gramática se han utilizado las restricciones identificadas en la definición del lenguaje de representación, y restricciones adicionales identificadas en el lenguaje utilizado en las descripciones. La inferencia de las reglas de la gramática se ha visto facilitada por la definición del dominio (i.e. las descripciones de componentes) desde el comienzo del desarrollo del sistema. Esta gramática se utiliza también para el procesamiento de las consultas.

Se han desarrollado una serie de experimentos para el estudio de la efectividad del sistema. En estos experimentos se ha utilizado un conjunto de consultas de prueba de 22 elementos. La gramática utilizada ha permitido procesar correctamente el 83% de las consultas. En los experimentos, el sistema presenta un incremento en la efectividad relativo del 21.5% en la *precision* para *recall* = 1 respecto a una aproximación basada en el modelo del espacio vectorial, uso de pesos de términos, extracción de raíces y listas de parada. Así mismo, la utilización de la gramática de unificación, con información lingüística específica del dominio, se presenta como decisiva para esta mejora en la efectividad respecto a una aproximación basada exclusivamente en la utilización de la información léxica de propósito general de WordNet.

CAPITULO 6

CONCLUSIONES Y FUTUROS DESARROLLOS

6.1 Principales aportaciones

En esta tesis hemos realizado una revisión crítica de los principales aspectos del proceso de Recuperación de Información, especialmente los relacionados con su efectividad. Así mismo hemos analizado los modelos y técnicas utilizados para el procesamiento de textos y consultas en este tipo de sistemas. Como consecuencia de nuestro análisis, nos hemos centrado en el estudio del modelo del espacio vectorial, la utilización de pesos de términos, listas de parada y algoritmos de extracción de raíces.

Se ha desarrollado un sistema de ayuda para la utilización de bibliotecas de componentes software en el que las técnicas de Recuperación de Información juegan un papel fundamental. El sistema Argos se ha desarrollado como sistema de ayuda para el conjunto de órdenes del sistema operativo UNIX y procesa el manual existente en formato electrónico en este entorno. Aunque el sistema se ha desarrollado para esta colección de componentes, su aplicación a otras diferentes es inmediata. Argos incluye una serie de elementos y funcionalidades, tales como el modelado del usuario, la realimentación, la especificación de consultas en lenguaje natural y funciones de navegación basadas en hipertexto, orientadas a proporcionar ayuda de la forma más efectiva a los usuarios de las bibliotecas de componentes. El sistema incluye un módulo que encapsula las funcionalidades más directamente relacionadas con las técnicas de Recuperación de Información antes citadas.

En nuestro estudio se ha hecho patente una importante evidencia experimental de que los sistemas basados en estas técnicas proporcionan una efectividad en el proceso de recuperación difícil de superar por otras aproximaciones.

No obstante, y con el fin de mejorar la efectividad, se ha realizado un estudio profundo de la utilización de técnicas de Procesamiento del Lenguaje Natural para la Recuperación de Información. Se han estudiado los sistemas de PLN que incluyen funcionalidades más próximas a la RI: los interfaces en lenguaje natural, los sistemas de comprensión de texto y los sistemas de extracción de información. Se han analizado los sistemas de RI que utilizan diversas técnicas de PLN y se han

diferenciado dos aproximaciones: la basada en la sintaxis y la basada en la semántica. En nuestro estudio, se ha hecho patente que, aunque aún no existe una definición clara de modelo para la aproximación basada en la semántica, los sistemas que siguen esta aproximación introducen mejoras en la efectividad del proceso de recuperación superiores a los basados en la sintaxis. La profundidad del análisis realizado por los diferentes sistemas que siguen una aproximación basada en la semántica varía de unos otros. Dos problemas comunes a todos los sistemas basados en la semántica son su especificidad de dominio y su importante esfuerzo de desarrollo.

La principal aportación de esta tesis consiste en la propuesta de un nuevo modelo de sistema de ayuda para la utilización de bibliotecas de componentes software centrado en el procesamiento de la documentación en lenguaje natural existente en estos entornos. En este modelo se integran técnicas de PLN en la RI, siguiendo una aproximación basada en la semántica. Para materializar este modelo, se ha diseñado e implementado el sistema Ares. El sistema se especializa en los problemas que plantea la brevedad de las descripciones de un número importante de colecciones de componentes software. Ares procesa el conjunto de las 432 descripciones cortas de las órdenes de UNIX de la sección 1 del manual del sistema operativo. Este conjunto de descripciones se ha tomado como representante suficientemente significativo del caso de las descripciones cortas de componentes software. Si bien Ares es específico de este dominio, resulta adaptable de forma directa a otras bibliotecas de componentes, como de hecho ya se ha realizado para la biblioteca de SmallTalk. El sistema se ha diseñado de forma que se mejore la efectividad del proceso de recuperación, se disminuya el esfuerzo de desarrollo y se facilite su adaptación a otras colecciones de componentes software.

Para reducir el esfuerzo de desarrollo, el léxico del sistema Ares se ha construido de forma automática a partir de la información existente en una base de datos léxica, WordNet. También se ha definido una forma de representación semántica de las descripciones, basada en los roles de las gramáticas de casos y los significados de los términos originalmente definidos en WordNet. Esta forma de representación se adecua tanto al lenguaje utilizado en las descripciones como a los objetos descritos (i.e. las órdenes de UNIX). Para la implementación del analizador-traductor, se ha desarrollado de forma incremental una gramática de unificación que permite procesar correctamente el 64.8% de las descripciones cortas del manual. La gramática así desarrollada se utiliza también para el procesamiento de las consultas de los usuarios. El método de cálculo de la similitud entre descripciones y consultas se ha definido de forma que se considera la estructura semántica de las expresiones y los significados de los términos que aparecen en ellas.

Finalmente, se ha desarrollado una serie de experimentos para el estudio de la efectividad del sistema. En los experimentos se ha utilizado una colección de 22 consultas de prueba para el conjunto de órdenes de UNIX. Se han implementado dos sistemas adicionales para la comparación de su comportamiento con el de Ares. El primero de estos sistemas implementa una aproximación basada en el modelo del espacio vectorial, pesos de términos, extracción de raíces y lista de parada, adaptada al caso de las descripciones cortas. El segundo implementa esta aproximación, utilizando además la información léxica incluida en Ares. En los

experimentos se hace patente el incremento en la efectividad conseguida por Ares respecto a la aproximación basada en el modelo del espacio vectorial y el papel determinante que juega la representación basada en la estructura semántica de las expresiones respecto a la aproximación basada en la información léxica exclusivamente.

6.2 Futuros desarrollos

Las posibles vías de continuación de este trabajo se refieren en su mayor parte a la investigación sobre la integración de diferentes técnicas de PLN en la RI, en el dominio de la documentación de bibliotecas de componentes software y en sistemas informáticos más generales.

En primer lugar, nos interesa el estudio de la adaptación del modelo propuesto en Ares a colecciones de componentes diferentes de las órdenes de UNIX trabajo que ya ha comenzado con el desarrollo de un analizador para la documentación del entorno de SmallTalk VisualWorks. Nuestro objetivo es profundizar en el proceso de adquisición del conocimiento del dominio para facilitar la confección de la gramática del analizador, investigando hasta dónde es posible automatizar este proceso. Además, la posibilidad de disponer de otros sistemas basados en el modelo propuesto en Ares, aplicados a diferentes bibliotecas de componentes software nos permitiría hacer un estudio más detallado de la efectividad de esta aproximación. Para este estudio de la efectividad se necesitaría también confeccionar conjuntos de consultas específicos de cada biblioteca de componentes considerada. Estos conjuntos de consultas podrían obtenerse directamente de las formuladas por conjuntos de usuarios a los que se les proporcionase un entorno de ayuda semejante a Argos, adaptado a la biblioteca específica. Esta investigación se encontraría, en principio, centrada en la utilización del modelo propuesto en Ares para la resolución de los problemas planteados por la brevedad de las descripciones.

Por otra parte, en otras bibliotecas de componentes el problema que se presenta es el opuesto al considerado en Ares. Existen bibliotecas de componentes y entornos informáticos que incluyen descripciones que plantean problemas precisamente por su excesiva longitud. Un ejemplo lo constituye la misma sección 1 del manual de UNIX si consideramos las descripciones completas de las órdenes. Cuando los documentos son de excesiva longitud, la presentación de éstos, de forma íntegra, desorienta y sobrecarga de trabajo al usuario. En este sentido, la implementación de operaciones sobre los documentos, como la segmentación de textos, o la categorización de textos, puede conllevar mejoras importantes en la efectividad del proceso de recuperación. La investigación encaminada en este sentido se centraría en la utilización de técnicas de PLN para la implementación de estas operaciones sobre los documentos. Estas técnicas pueden ser diferentes de las utilizadas en Ares, con un análisis más próximo al de tipo "text-skimming", o, en contraposición, potenciando los criterios estadísticos y la información léxica, dependiendo de los casos, y comparando la efectividad de las diferentes aproximaciones.

Otro tema importante es el de la integración de todo este conjunto de operaciones sobre documentos y consultas en lenguaje natural, en entornos de ayuda a la

utilización de bibliotecas de componentes, en los que se incluya un número de funcionalidades adicionales a las definidas en Argos. Estas funcionalidades se pueden encontrar encaminadas a facilitar a los usuarios operaciones sobre la biblioteca de componentes diferentes de la de recuperación, como la adaptación o la integración. De esta forma, interesa estudiar el papel, o los papeles, que pueden jugar los métodos de análisis de textos en la ayuda a la realización de estas otras operaciones de los usuarios sobre las bibliotecas de componentes. La continuación de esta línea consistiría en la investigación de la adaptación de las operaciones sobre textos y consultas, en sistemas de ayuda para otros entornos informáticos diferentes de las colecciones de componentes software.

Finalmente, a más largo plazo, y a partir de la experiencia adquirida en el desarrollo de sistemas como los anteriores, resulta interesante la definición de una metodología para el desarrollo de sistemas de RI basados en el PLN. En esta metodología, los primeros pasos consistirían en la definición del dominio y la funcionalidad precisa del sistema, así como el análisis de costes de desarrollo de cada una de las diferentes alternativas basadas en distintos criterios. En este análisis se identificarían las características de los textos que aconsejan o desaconsejan la utilización de las distintas técnicas estudiadas.

APENDICE

DETALLE DE IMPLEMENTACION DEL SISTEMA ARES Y DATOS EXPERIMENTALES OBTENIDOS

En este apéndice se presentan los principales detalles de implementación del sistema Ares, así como los correspondientes a los sistemas utilizados y los valores de similitud obtenidos en los experimentados presentados en el capítulo 5. La organización general del sistema se describe en el capítulo 5, y se encuentra de acuerdo con el Diagrama de Flujo de Datos de la figura 5.6.

Se presenta el código correspondiente a los módulos de Ares y los sistemas utilizados en los experimentos (A.1). A continuación, los preprocesadores desarrollados para el tratamiento de la documentación del manual existente en el S.O. y representación en sintaxis de hechos Prolog, y la obtención del léxico del sistema (A.2). El conjunto de hechos Prolog que constituye el léxico del sistema se encuentra en (A.3). En (A.4) se incluyen los principales elementos de datos de entrada del núcleo de Ares y de sistemas utilizados en los experimentos, formados por las descripciones de las órdenes y los hechos correspondientes a las consultas. Finalmente, se presentan los valores de similitud de las descripciones a las consultas calculados por los sistemas en los experimentos (A.5).

A.1 Núcleo de Ares y módulos utilizados en los experimentos

Los módulos para la implementación de las principales funcionalidades del sistema:

1. Análisis de documentos y consultas
2. Cálculo de similitud
3. Sistemas MV y MV-S
4. Predicados para las pruebas
5. Utilidades
6. Compatibilidad

se han desarrollado en Prolog y su código se incluye a continuación.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%    1. ANALISIS DE DOCUMENTOS Y CONSULTAS
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%
%% gramatica
%% =========
%%
%% el predicado utilizado para el análisis:
%%
%%    sentences(CaseFrameList,Descripcion,[]).
%%
%% toma como entrada la expresión de entrada (Descripción) y proporciona
%% como resultado del análisis su representación en forma de frame
%% de casos.
%%

%%
%% reglas de formación de oraciones
%%
%% el resultado es el frame de casos representando las oraciones
%%

sentences([CaseFrame]) -->
      sentence(CaseFrame).
sentences([CaseFrame1, CaseFrame2]) -->
      sentence(CaseFrame1),
      coord,
      sentence(CaseFrame2).

sentence([ActionSlot,ObjectSlot|SlotList]) -->
      verbs(ActionFiller),
      noun_phrases(ObjectFiller),
      ( cases(SlotList); [], ( SlotList = [] } ),
      { ActionSlot = [action,ActionFiller],
        ObjectSlot = [object,ObjectFiller] }.

%%
%% reglas de formación de verbos
%%
%% el resultado del análisis de la parte de la frase correspondiente
%% al verbo es un slot en el que el valor de la cabeza (head) es el
%% termconcept del verbo
%%

verbs( [ [head([VerbTermConcept]),modif([])] ]) -->
      verb(VerbTermConcept).
verbs( [ [head([VerbTermConcept]),modif([])] | VerbTermConceptL]) -->
      verb(VerbTermConcept),
      coord,
      verbs(VerbTermConceptL).

verb(termconcept(Verb,ConceptL)) -->
      [Word],
      { morphie(Word,Verb,verb),
        word([Verb],verb,ConceptL) }.
verb(termconcept(Verb,ConceptL)) -->
      [Word,Prep],
      { morphie(Word,Verb,verb),
        word([Verb,Prep],verb,ConceptL) }.

%%
%% reglas de formación de casos
%%
%% el resultado es la lista de slots con roles semanticos correspondiente
%% a los casos de las frases nominales
```

```
%%

cases([Slot]) -->
      case(Slot).
cases([Slot|SlotList]) -->
      case(Slot),
      cases(SlotList).
case([Role,Filler]) -->
      case_marker(RoleList),
      noun_phrases(Filler),
      { member(Role,RoleList) }.

%%
%% reglas de formación de frases nominales
%%
%% el resultado es el filler del slot correspondiente a la frase, con
%% estructura [head,modif]
%%

noun_phrases([NounPhraseAn]) -->
      noun_phrase(NounPhraseAn).
noun_phrases([NounPhraseAn|NounPhraseAnL]) -->
      noun_phrase(NounPhraseAn),
      coord,
      noun_phrases(NounPhraseAnL).

noun_phrase([head(HeadList),modif(ModifList)]) -->
      ( det;
        []),
      ( qualifiers(ModifList1);
        [],
        { ModifList1 = []}
      ),
      np_head(HeadList),
      ( complement(ModifList2);
        [],
        { ModifList2 = []} ),
      { append(ModifList1,ModifList2,ModifList) }.

%%
%% regla de formación del núcleo de frases nominales
%%
%% el resultado es la representación del núcleo mediante su
%% terminoconcepto asociado
%%

np_head([termconcept(Head,ConceptList)]) -->
      [Word],
      { morphie(Word,Head,noun),
        word([Head],noun,ConceptList) }.

%%
%% reglas de formación de calificadores
%%
%% el resultado es la representación de los calificadores mediante
%% sus terminoconceptos asociados
%%

qualifiers([Qualifier]) -->
      qualifier(Qualifier).
qualifiers([Qualifier|QualifierList]) -->
      qualifier(Qualifier),
      qualifiers(QualifierList).
qualifier(termconcept(Qualifier,ConceptList)) -->
      [Word],
      { morphie(Word,Qualifier,noun),
          word([Qualifier],noun,ConceptList);
        morphie(Word,Qualifier,adje),
        word([Qualifier],adje,ConceptList) }.

%%
%% reglas de formación de complementos nominales
%%
%% el resultado es la representación de los complementos mediante
```

```
%% sus terminoconceptos asociados
%%

complement(ComplementList) -->
      ( [of];
        [about] ),
      noun_phrase([head(HeadList),modif(ModifList)]),
      { append(HeadList,ModifList,ComplementList) }.

%%
%% reglas de formación de marcadores de casos
%%
%% se obtiene el role semántico indicado por los asociados a
%% a las preposiciones o expresiones preposicionales
%%

case_marker(Role) -->
      [Word],
      { prepcv(Word,Role) }.
case_marker(Role) -->
      [Word1,Word2],
      { exprcv([Word1,Word2],Role) }.

%%
%% reglas de formación de elementos sencillos
%%

coord -->
      [Word], { conj(Word)}.
coord -->
      [Word], { conj(Word)}, coord.
det -->
      [Word], { art(Word) }.
enlace -->
      [Word], { conj(Word) }.

%%
%% preposiciones con roles tematicos asociados
%%

prepcv(to,[destination,purpose]).
prepcv(by,[instrument,manner]).
prepcv(for,[purpose,duration]).
prepcv(in,[source,destination,manner,time]).
prepcv(on,[source,destination,time]).
prepcv(from,[source,time]).
prepcv(at,[manner,time]).
prepcv(between,[comparison]).
prepcv(as,[comparison,manner]).
prepcv(with,[instrument,manner]).
prepcv(without,[instrument,manner]).
prepcv(when,[time]).
prepcv(until,[time]).
prepcv(into,[destination]).
prepcv(via,[instrument,manner]).
exprcv([according,to],manner).

%%
%% articulos y conjunciones
%%

art(a).
art(an).
art(the).

art(your).
art(my).
art(its).

conj(',').
conj(';').
conj(and).
conj(or).
conj(either).
```

```
conj(neither).

%%
%% analizador morfologico
%% ======================
%%
%% el predicado de entrada para el analisis es
%%
%%    morphie(Palabra,Termino,CategoriaSintactica)
%%
%% dada una palabra en sus diferentes formas (Completo), se obtiene su
%% representación canonica (Termino) y la categoría sintáctica asociada
%%

morphie(Completo,Termino,Categoria) :-
      explode(Completo,ComplL),
      morf(Categoria, Raiz, Terminacion, ComplL, []),
      append(Raiz,Terminacion,TerminoL),
      explode(Termino,TerminoL),
      word([Termino],Categoria,_).

%%
%% reglas de formación de nombres
%%

morf(noun, R, []) --> root(R), [s].
morf(noun, R, [s]) --> root(R), [s,e,s].
morf(noun, R, [x]) --> root(R), [x,e,s].
morf(noun, R, [z]) --> root(R), [z,e,s].
morf(noun, R, [c,h]) --> root(R), [c,h,e,s].
morf(noun, R, [s,h]) --> root(R), [s,h,e,s].
morf(noun, R, [y]) --> root(R), [i,e,s].
morf(noun, R, []) --> root(R).

%%
%% reglas de formación de verbos
%%

morf(verb, R, []) --> root(R), [s].
morf(verb, R, [y]) --> root(R), [i,e,s].
morf(verb, R, [e]) --> root(R), [e,s].
morf(verb, R, []) --> root(R), [e,s].
morf(verb, R, [e]) --> root(R), [e,d].
morf(verb, R, []) --> root(R), [e,d].
morf(verb, R, [e]) --> root(R), [i,n,g].
morf(verb, R, []) --> root(R), [i,n,g].
morf(verb, R, []) --> root(R).

%%
%% reglas de formación de adjetivos
%%

morf(adje, R, []) --> root(R), [e,r].
morf(adje, R, []) --> root(R), [e,s,t].
morf(adje, R, [e]) --> root(R), [e,r].
morf(adje, R, [e]) --> root(R), [e,s,t].
morf(adje, R, []) --> root(R), [].

%% regla de obtención de raices para parejas de terminos

root([]) --> [].
root([H|Q]) --> [H], root(Q).

%%
%% analisis de descripciones
%% =========================
%%
%% el predicado de entrada:
%%
%%    analizar_manual
%%
%% es utilizado para realizar el analisis de todo el manual, se ocupa de
%% almacenar los resultados de los análisis de las descripciones en
%% la base de datos en el hecho command/3.
```

```
%%

analizar_manual :-
      retractall(command(_,_,_)),
      analizar_docs.
analizar_docs :-
      command(name(Name),description(Description)),
      write('  '), write(analisis(Name)), nl,
      sentences(Analisis,Description,[]),
      assert(command(name(Name),description(Description),
            analisis(Analisis))),
      fail.
analizar_docs.

%%
%% analisis de consultas
%% =====================
%%
%% el predicado de entrada:
%%
%%    analizar_consultas
%%
%% es utilizado para realizar el analisis del conjunto de consultas,
%% se ocupa de almacenar los resultados de los análisis de las consultas en
%% la base de datos en el hecho query/3.
%%

analizar_consultas :-
      retractall(query(_,_,_)),
      analizar_cons.
analizar_cons :-
      query(reference(Ref),words(Description)),
      write(analisis(Ref)), nl,
      sentences(Analisis,Description,[]),
      assert(query(reference(Ref),words(Description),analisis(Analisis))),
      fail.
analizar_cons.

%%
%% calculo del peso de los terminos basado en idf
%% ==============================================
%%
%% el predicado de entrada
%%
%%    generate_wordw
%%
%% se ocupa de generar los pesos de todos los términos de acuerdo con su
%% frecuencia de aparición en las descripciones, el resultado del analisis
%% se almacena en la base de datos en el hecho wordw/2
%%

generate_wordw :-
      retractall(wordw(_,_)),
      setof(Word,word(Word),WordList),
      setof(Document,doc_name(Document),DocumentList),
      length(DocumentList,NumDocs),
      assertWordsWeight(WordList,NumDocs).

assertWordsWeight([],_).
assertWordsWeight([Word|WordL],NumDocs) :-
      df(Word,Df),
      %% calculo de pesos mediante la expresion del
      %% logaritmo
      Weigth is log( NumDocs / Df) / 0.693147,
      assert(wordw(Word,Weigth)),
      write(wordw(Word,Weigth)), nl,
      assertWordsWeight(WordL,NumDocs).

%%
%% df: calculo de df (document frequency) del termino
%%

df(Termino,Df) :-
      setof(DocName,doc(Termino,DocName),DocNameList), !,
```

```
        length(DocNameList,Df).
df(_,1).

%%
%% otros predicados
%%

doc_name(DocName) :-
        command(DocName,_).
doc(Term,DocName) :-
        command(name(DocName),description(WordList)),
        member(Word,WordList),
        term(Word,Term).
term(Word,Term) :-
        morphie(Word,Term,_), !.
word(Word) :-
        word([Word],_,_).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%      2. CALCULO SIMILITUD SEMANTICA
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%
%% calculo de la similitud (o relevancia)
%% ====================================
%%
%% el predicado de acceso es:
%%
%%      relevancia(ReferenciaDeQuery,ListaDeDocumentosRelevantes)
%%
%% dada la referencia de una consulta, calcula la similitud de los documentos
%% existentes en la base de datos y genera la lista de aquellos con similitud
%% mayor que cero y ordenados decrecientemente por este valor
%%

relevancia(RefQuery,DocRelList) :-
        setof(Doc,similitud_qd(RefQuery,Doc),UnOrderedDocRelList),
        relsort(UnOrderedDocRelList,DocRelList).

%%
%% similitud entre frames de consulta y documento
%%

similitud_qd(RefQ,doc(NameD,Sim)) :-
        command(name(NameD),_),
        setof(S,similitud_qd0(RefQ,NameD,S),SimL),
        max(SimL,0,Sim).
similitud_qd0(RefQ,NameD,Sim) :-
        query(reference(RefQ),_,analisis(FramesQ)),
        command(name(NameD),_,analisis(FramesD)),
        sim_frames_qd(FramesQ,FramesD,Sim).

sim_frames_qd([FrameQ],[FrameD],Sim) :-
        sim_frame(FrameQ,FrameD,Sim).
sim_frames_qd([FrameQ],[FrameD1,FrameD2],Sim) :-
        sim_frame(FrameQ,FrameD1,Sim1),
        sim_frame(FrameQ,FrameD2,Sim2),
        Sim is Sim1+Sim2.

%%
%% similitudes entre frames, slots, fillers y terminoconceptos
%%

sim_frame([],_,0) :- !.
sim_frame(_,[],0) :- !.
sim_frame(Frame1,Frame2,Sim) :-
        sim_slot(Frame1,Frame2,Frame1b,Frame2b,SimS),
        sim_frame(Frame1b,Frame2b,SimFr),
        Sim is SimS + SimFr.
```

```
sim_slot([Slot|Frame1b],Frame2,Frame1b,Frame2b,SimS) :-
      Slot = [Role,Filler11],
      ( take_member(Frame2,Frame2b,[Role,Filler12]), !,
        sim_filler1(Filler11,Filler12,SimS);
          SimS = 0).

sim_filler1([],_,0).
sim_filler1([Filler|Filler11],Filler12,SimF1):-
      sim_filler1a(Filler,Filler12,SimF1a),
      sim_filler1(Filler11,Filler12,SimF1b),
      SimF1 is SimF1a + SimF1b.

sim_filler1a(_,[],0).
sim_filler1a(Filler1,[Filler2|Filler12],SimF1a) :-
      Filler1 = [head(Head1TCL),modif(Modif1TCL)],
      Filler2 = [head(Head2TCL),modif(Modif2TCL)],
      sim_TermConceptList(Head1TCL,Head2TCL,SH),
      sim_TermConceptList(Modif1TCL,Modif2TCL,SM),
      sim_filler1a(Filler1,Filler12,SimF1a2),
      SimF1a is SH + SM + SimF1a2.

sim_TermConceptList(TermConceptL1,TermConceptL2,Sim) :-
      setof(Val,sim_TCLL(TermConceptL1,TermConceptL2,Val),ValL),
      max(ValL,0,Sim).

sim_TCLL([],_,0).
sim_TCLL(_,[],0).
sim_TCLL(TermConceptL1,TermConceptL2,Sim) :-
      member(TermConcept1,TermConceptL1),
      member(TermConcept2,TermConceptL2),
      sim_TC(TermConcept1,TermConcept2,Sim).

sim_TC(TermConcept1,TermConcept2,Sim) :-
      TermConcept1 = termconcept(Term1,ConceptL1),
      TermConcept2 = termconcept(Term2,ConceptL2),
      member(Concept,ConceptL1),
      member(Concept,ConceptL2),
      wordw(Term1,WordWeight1),
      wordw(Term2,WordWeight2),
      Sim is WordWeight1 * WordWeight2.
sim_TC(_,_,0).

take_member(ListIn,ListOut,Member) :-
      append(A,[Member|B],ListIn),
      append(A,B,ListOut).

max([],Max,Max).
max([A|B],Val,Max):-
      A > Val, !, max(B,A,Max).
max([_|B],Val,Max):-
      max(B,Val,Max).

%%
%% ordenacion por relevancia, por orden alfabetico
%% y eliminacion de documentos con rel nula
%%

relsort(LIn,LOut) :-
      remove_zero(LIn,L1),
      alpha_sort(L1,L2),
      rel_sort(L2,LOut).

remove_zero([],[]).
remove_zero([H|T],L) :-
      H = doc(_,0), !, remove_zero(T,L).
remove_zero([H|T],[H|L]) :-
      remove_zero(T,L).

alpha_sort([],[]).
alpha_sort([H|T],S) :-
      alpha_sort(T,A), alpha_insert(H,A,S).
alpha_insert(E,[],[E]).
alpha_insert(E,[H|T],[E,H|T]) :-
```

```
        E = doc(NE,_), H = doc(NH,_), strcmp(>,NE,NH), !.
alpha_insert(E,[H|T],[H|S]) :-
        alpha_insert(E,T,S).

rel_sort([],[]).
rel_sort([H|T],S) :-
        rel_sort(T,A), insert(H,A,S).
insert(E,[],[E]).
insert(E,[H|T],[E,H|T]) :-
        E = doc(_,RE), H = doc(_,RH), RE > RH, !.
insert(E,[H|T],[H|S]) :-
        insert(E,T,S).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%     3. SISTEMAS MV Y MVS
%%
%%         MV:     mod. vectorial, pesos, stoplists y stemming
%%         MVS:    MV + sinonimia
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%% calculo de la similitud (o relevancia)
%% ======================================
%%
%% los predicados de acceso son:
%%
%%     MV:  relevancia_bp(ReferenciaDeQuery,ListaDeDocumentosRelevantes)
%%     MVS: relevancia_th(ReferenciaDeQuery,ListaDeDocumentosRelevantes)
%%
%% dada la referencia de una consulta, calculan la similitud de los documentos
%% existentes en la base de datos y genera la lista de aquellos con similitud
%% mayor que cero y ordenados decrecientemente por este valor
%%
%% MV:
%%     - extracción de raíces (stemming) se realiza mediante el análisis
morfológico
%%       implementado por el predicado morphie
%%     - lista de palabras vacías se utiliza la incluida en el sistema
%%       ORBIT (8 palabras)
%%     - Pesos de las palabras almacenados en el hecho wordw que proporciona
%%       el peso de los terminos basado en el idf calculado en el analisis
%%       inicial del manual
%% MVS:
%%     - Elementos de MV mas información de sinonimia existente en el lexico
%%

relevancia_bp(RefQ,DocList) :-
        setof(D,similitud_qd_frec(RefQ,D,bp),Doc),
        relsort(Doc,DocList).

relevancia_th(RefQ,DocList) :-
        setof(D,similitud_qd_frec(RefQ,D,th),Doc),
        relsort(Doc,DocList).

%%
%% calculo de la similitud por los dos metodos
%% ==========================================
%%
%% similitud_qd_frec(ReferenciaDeQuery,Documento,MétodoDeCalculo)
%%     con método de cálculo
%%           MV  <=>  MetodoDeCalculo = bp
%%           MVS <=>  MetodoDeCalculo = th
%%

similitud_qd_frec(RefQ,doc(NameD,Sim),Method) :-
        query(reference(RefQ),words(QWordL)),
        command(name(NameD),description(DWordL)),
        similitud_frec(QWordL,DWordL,Sim,Method).
```

```
similitud_frec(QWordL,DWordL,Similitud,Method) :-
      wordl_stem(QWordL,QStemL),
      wordl_stem(DWordL,DStemL),
      sim_wordl1(QStemL,DStemL,Sim,Method),
      ( Sim = 0, Similitud = 0, !;
        vector_module2(QStemL,QVM),
        vector_module2(DStemL,DVM),
        Similitud is Sim / sqrt(QVM * DVM) ).

sim_wordl1([],_,0,_).
sim_wordl1([H|T],DWordL,Sim,Method) :-
      sim_wordl(H,DWordL,S1,Method),
      sim_wordl1(T,DWordL,S2,Method),
      Sim is S1+S2.

sim_wordl(_,[],0,_).
sim_wordl(Word,[H|T],Sim,Method) :-
      sim_word(Word,H,S1,Method),
      sim_wordl(Word,T,S2,Method),
      Sim is S1+S2.

sim_word(W,W,Sim,_) :-
      wordw(W,WW), Sim is WW * WW, !.
sim_word(W1,W2,Sim,th) :-
      word([W1],CS,SynSets1), word([W2],CS,SynSets2),
      member(Syn,SynSets1), member(Syn,SynSets2),
      wordw(W1,WW1), wordw(W2,WW2), !, Sim is WW1 * WW2.
sim_word(_,_,0,_).

vector_module2([],0).
vector_module2([Word|WordL],VM) :-
      wordw(Word,WW), vector_module2(WordL,VML), VM is WW * WW + VML.

wordl_stem([],[]).
wordl_stem([Word|WordL],StemL) :-
      stoplist(Word), !, wordl_stem(WordL,StemL).
wordl_stem([Word|WordL],[Stem|StemL]) :-
      morphie(Word,Stem,_), !, wordl_stem(WordL,StemL).
wordl_stem([_|WordL],StemL) :-
      wordl_stem(WordL,StemL).

stoplist(and).
stoplist(a).
stoplist(an).
stoplist(by).
stoplist(from).
stoplist(of).
stoplist(the).
stoplist(with).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%    4. PREDICADOS PARA LAS PRUEBAS
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Predicados para:
%%
%%    - realizar el analisis completo del manual y de las consultas y
%%      almacenarlos en archivos en disco.
%%    - cargar los analisis desde disco si ya se han realizado
%%    - obtener relevancias de los documentos para la bateria de consultas y
%%      almacenarlos en disco
%%    - imprimir ranking de documentos
%%    - estadisticas de documentos recuperados y relevantes, calculo de
%%      valores de recall y precision

%%
%% analizar manual y consultas y grabarlo en disco
%% ==========================================
%%
```

```
analizar_a_disco :-
      cargar_datos_iniciales,
      analizar,
      write('  grabando resultados...'), nl,
      tell('analisisdescrip.pro'),
      listing(command/3),
      listing(wordw/2),
      told,
      tell('analisisconsultas.pro'),
      listing(query/3),
      told.
analizar :-
      analizar_manual,
      generate_wordw,
      analizar_consultas.

%%
%% cargar analisis ya realizados en disco
%% ======================================
%%

cargar_datos :-
      cargar_datos_iniciales,
      write('  carga del analisis realizado del manual...'), nl,
      reconsult('analisisdescrip.pro'),
      write('  carga del analisis realizado de las consultas...'), nl,
      reconsult('analisisconsultas.pro').
cargar_datos_iniciales :-
      write('  carga de manual, lexico y consultas...'), nl,
      reconsult('man.pro'),
      reconsult('lexico.pro'),
      reconsult('consultas.pro').

%%
%% obtener relevancias de documentos para la bateria de consultas
%% ==============================================================
%%

%%
%% relevancia semantica
%%

calculo_relevancias_sem :-
      retractall(query_doc_rel_sem(_,_)),
      calc_rel_sem,
      tell('relevanciasem.pro'),
      listing(query_doc_rel_sem),
      told.
calc_rel_sem :-
      query(reference(R),_),
      write(query(R)), nl,
      relevancia(R,DL),
      assert(query_doc_rel_sem(reference(R),docRel(DL))),
      write(docrel(DL)), nl,
      fail.
calc_rel_sem.

%%
%% relevancia sistema MV
%%

calculo_relevancias_bp :-
      retractall(query_doc_rel_bp(_,_)),
      calc_rel_bp,
      tell('relevanciasbp.pro'),
      listing(query_doc_rel_bp),
      told.
calc_rel_bp :-
      query(reference(R),_),
      write(query_bp(R)), nl,
      relevancia_bp(R,DL),
      assert(query_doc_rel_bp(reference(R),docRel(DL))),
      fail.
calc_rel_bp.
```

```
%%
%% relevancia sistema MVS
%%

calculo_relevancias_th :-
      retractall(query_doc_rel_th(_,_)),
      calc_rel_th,
      tell('relevanciasth.pro'),
      listing(query_doc_rel_th),
      told.
calc_rel_th :-
      query(reference(R),_),
      write(query_th(R)), nl,
      relevancia_th(R,DL),
      assert(query_doc_rel_th(reference(R),docRel(DL))),
      fail.
calc_rel_th.

%%
%% ranking
%% =======
%%
%% manejo de la base de datos con valores de similitudes
%% para ordenación e impresión por relevacia (ranking)
%%

print_ranking :-
      write('ARES:'), nl, print_docs_rel(ares), nl, fail;
      write('MV:'), nl, print_docs_rel(mv), nl, fail;
      write('MVS:'), nl, print_docs_rel(mvs), nl, fail;
      true.

print_docs_rel(Method) :-
      ( Method = ares, query_doc_rel_sem(reference(Reference),docRel(DocL));
        Method = mv, query_doc_rel_bp(reference(Reference),docRel(DocL));
        Method = mvs, query_doc_rel_th(reference(Reference),docRel(DocL)) ),
      bagof(DocName,docname(DocName,DocL),DocNameL),
      write(ref(Reference)), write(doc(DocNameL)), nl, fail.

print_doc_rel_ref(Reference,DocL) :-
      bagof(DocName,docname(DocName,DocL),DocNameL),
      write(ref(Reference)), write(doc(DocNameL)), nl.
docname(DN,DL) :-
      member(doc(DN,_),DL).

%%
%% calculos orientados a la obtención de recall y precision
%% ========================================================
%%

print_recalls(Method) :-
      pos_rel(Method,Reference,NRels,NRets,NRelRets,PosRel),
      ( PosRel = [Pos1|_], Precision is 1 / Pos1;
        PosRel = [], Precision = 0),
      write(Reference), tab(5), write(NRels), tab(5), write(NRets), tab(5),
      write(NRelRets), tab(5), write(PosRel), tab(5), write(Precision), nl,
      fail.
print_recalls(_).

pos_rel(Method,Reference,NRels,NRets,NRelRets,PosRel) :-
      query_rel(reference(Reference),docs(DocRelL) ),
      ( Method = ares, query_doc_rel_sem(reference(Reference),docRel(DocL));
        Method = mv, query_doc_rel_bp(reference(Reference),docRel(DocL));
        Method = mvs, query_doc_rel_th(reference(Reference),docRel(DocL)) ),

      bagof(DocName,docname(DocName,DocL),DocNameL),
      positions(DocRelL,DocNameL,0,PosRel),
      length(DocRelL,NRels),
      length(DocNameL,NRets),
      length(PosRel,NRelRets).

positions(_,[],_,[]).
positions(DocRelL,[DN|DNL],Pos,PosL) :-
```

```
      Pos1 is Pos+1,
      ( member(DN,DocRelL),
        PosL = [Pos1|PosL1], !;
        PosL = PosL1 ),
      positions(DocRelL,DNL,Pos1,PosL1).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%    5. UTILIDADES
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%% Utilidades para estadísticas de documentos, consultas y léxico
%% ==============================================================
%%

%%
%% numero de documentos en la base de datos
%%

document_num(Num) :-
      bagof(DN,name_doc(DN),DNL), length(DNL,Num).
name_doc(DN) :-
      command(DN,_).

%%
%% numero de palabras diferentes en descripciones
%%

term_num(Num) :-
      setof(W,term_doc(W),WL), length(WL,Num).
term_doc(Term) :-
      command(_,description(WL)), member(W,WL), term(W,Term).

%%
%% longitud media de las descripciones
%%

med_length(MLength) :-
      bagof(L,doc_length(L),LL),
      length(LL,Num), sum(LL,SLength),
      MLength is SLength / Num.
doc_length(L) :-
      command(_,description(WL)), length(WL,L).
sum([],0).
sum([H|T],S) :-
      sum(T,S1), S is S1 + H.

%%
%% numero de palabras diferentes en consultas
%%

term_num_cons(Num) :-
      setof(W,term_cons(W),WL), length(WL,Num).
term_cons(Term) :-
      query(_,words(WL)), member(W,WL), term(W,Term).
%%
%% longitud media de las consultas
%%

med_length_cons(MLength) :-
      bagof(L,cons_length(L),LL), length(LL,Num), sum(LL,SLength),
      MLength is SLength / Num.
cons_length(L) :-
      query(_,words(WL)), length(WL,L).

%%
%% estadisticas del lexico
%%

word_term_num(Words,Terms) :-
```

```
      bagof(W,wordw(W),WL), length(WL,Words),
      setof(T,wordw(T),TL), length(TL,Terms).
wordw(W) :-
      word([W],_,_).

means_num(Means) :-
      setof(M,mean(M),ML), length(ML,Means).
mean(M) :-
      word(_,_,MeanL), member(M,MeanL).

%%
%% Miscelanea
%% ==========
%%

%%
%% llamadas a predicados de uso frecuente (copy&paste)
%%

%% :- set_current_directory('Pandora:Desk:Aresv4:runtime').
%% :- set_current_directory('Dafne:runtime').
a :- reconsult('analizador.pro').

%%
%% Utilidades E/S
%%

writeL([]).
writeL([H|L]) :-
      write(H), nl, writeL(L).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%%     6. COMPATIBILIDAD
%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%
%% Los siguientes predicados son necesarios para la ejecución del programa
%% sobre SICStus Prolog
%%

%
 %  explode/2
 %
explode(Atom,CharList) :-
      nonvar(Atom), !,
      name(Atom,IntList),
      int_to_ascii(IntList,CharList).
explode(Atom,CharList) :-
      int_to_ascii(IntList,CharList),
      name(Atom,IntList).
int_to_ascii([],[]).
int_to_ascii([Int|IL],[Char|CL]) :-
      name(Char,[Int]),
      int_to_ascii(IL,CL).

/*
 * member/2
 */
member(_1, [_1|_2]).
member(_1, [_2|_3]) :-
    member(_1, _3).

/*
 * append/3
 */
append([], _1, _1).
append([_1|_2], _3, [_1|_4]) :-
    append(_2, _3, _4).

/*
```

```
 * strcmp/3
 */
strcmp(>,Atom1,Atom2) :-
      name(Atom1,A1L), name(Atom2,A2L),
      lcmp(A1L,A2L).
lcmp([],_) :- !, fail.
lcmp(_,[]) :- !.
lcmp([A1|_],[A2|_]) :- A1 < A2, !, fail.
lcmp([A1|_],[A2|_]) :- A1 > A2, !.
lcmp([_|L1],[_|L2]) :- lcmp(L1,L2).
```

A.2 Preprocesadores

En este punto se proporcionan los detalles de implementación de los preprocesadores con los que se obtiene el conjunto de hechos Prolog con las descripciones cortas de las órdenes de UNIX de la sección 1 del manual, y el léxico de Ares.

A.2.1 Preprocesamiento del manual

El programa *manaprolog.lex*, desarrollado en lex permite generar una base de datos en forma de hechos prolog a partir de la documentación incluida en manual de sistema operativo:

```
%START                                  descripcion finlinea finnombrecom
                                        nombrecomando

palabra                                 [A-Za-z]+
espacios                                [" "]+

%%

<finlinea>{espacios}                    { BEGIN nombrecomando;
                                          printf("documento(\n"); }
<finlinea>.                             { BEGIN 0; }

<nombrecomando>{palabra}                { BEGIN finnombrecom;
                                          printf("\tnombre('%s'),\n",yytext); }
<finnombrecom>[^")"]+")"{espacios}      { BEGIN descripcion;
                                          printf("\tdescripcion([");  }

<descripcion>{palabra}                  { printf("'%s'",yytext); }
<descripcion>", "                       { printf(",',',"); }
<descripcion>"(".")"                    ;
<descripcion>.                          { printf(","); }
<descripcion>\n                         { BEGIN finlinea; printf("]) ).\n"); }

.                                       ;
\n                                      { BEGIN finlinea; }
```

La obtención de la base de datos completa se realiza mediante la ejecución de la orden, desde el intérprete del S.O:

```
/> man 1 list | manaprolog > man1.pro
```

A.2.2 Obtención del léxico

La obtención del léxico del sistema a partir de la información existente en WordNet se realiza mediante los siguientes pasos:

1. Generación del archivo *lexico* con el conjunto de términos existente en las descripciones de las órdenes y las consultas (predicados command/2 y query/2, respectivamente, se describen en A.4) mediante el predicado en Prolog genera_lexico:

```
genera_lexico :-
      tell(lexico),
      genera_lexico(Lexico),
      writeL(Lexico),
      told.
genera_lexico(WordL) :-
      setof(Word,word_com_query(Word),WordL).
word_com_query(Word) :-
      command(_,description(DescriptionL)),
      member(Word,DescriptionL).
word_com_query(Word) :-
      query(_,words(DescriptionL)),
      member(Word,DescriptionL).
```

2. Se ejecuta el programa *generalex* que utiliza a su vez genera el léxico en sintáxis de prolog mediante la orden al S.O.:

```
      /> generalex < lexico
```

El programa *generalex.c* gestiona la consulta a WordNet y utiliza los programas *termino*.lex y wnaprol.*lex* que transforman los datos en el formato de WordNet a la sintáxis de Prolog. Se incluye a continuación el código de los tres.

generalex.c

```
/*
      Generalex.c
      Genera información léxica en sintaxis de prolog para una lista de palabras
*/

#include <stdio.h>

main()
{
      FILE  *file;
      char  palabra[100], comando[500];

      /* borrar ficheros auxiliares */

      system("rm terminos1  2> /dev/null");
      system("rm terminos2  2> /dev/null");
      system("rm lexico.pro 2> /dev/null");

      /* leer la lista de palabras obteniendo la lista de terminos (terminos1)*/

      printf("generando lista de terminos...\n");
      file = fopen("palabras","r");
      while( fscanf( file, "%s\n", palabra) == 1 ) {
            sprintf(comando, "wn %s | exec/termino >> terminos1", palabra);
            system(comando);
      }
      fclose(file);
```

```
        /* ordenar lista de terminos y eliminar duplicados */

        printf("ordenando terminos y eleminando duplicados...\n");
        system("sort terminos1 | uniq > terminos2");

        /* obtener informacion lexica representada en prolog */

        printf("generando lexico.pro...\n");
        file = fopen("terminos2","r");
        while(      fscanf( file, "%s\n", palabra) == 1 ) {
              sprintf(comando,
                    "look '%s ' /usr/local/lib/wordnet/index.noun |
                     exec/wnaprol >> lexico.pro",
                    palabra);
              system(comando);
              sprintf(comando,
                    "look '%s ' /usr/local/lib/wordnet/index.verb |
                     exec/wnaprol >> lexico.pro",
                    palabra);
              system(comando);
              sprintf(comando,
                    "look '%s ' /usr/local/lib/wordnet/index.adj |
                     exec/wnaprol >> lexico.pro",
                    palabra);
              system(comando);
              sprintf(comando,
                    "look '%s ' /usr/local/lib/wordnet/index.adv |
                     exec/wnaprol >> lexico.pro",
                    palabra);
              system(comando);
        }
        fclose(file);

        /* borrar ficheros auxiliares */

        system("rm terminos1  2> /dev/null");
        system("rm terminos2  2> /dev/null");
}
```

termino.lex

```
%START                          categoria termino
palabra                         [a-z]+

%%

"Information available for " BEGIN categoria;
<categoria>(palabra)          BEGIN termino;
<termino>" "                  ;
<termino>(palabra)            { printf("%s\n",yytext); BEGIN 0; }
\n                          |
.                           ;
%%
```

wnaprol.lex

```
%START                          pos fields syn

palabra                         [a-z]+
synset                          [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]

%%

<pos>" n "                      { printf("noun,["); BEGIN fields; }
<pos>" v "                      { printf("verb,["); BEGIN fields; }
<pos>" a "                      { printf("adje,["); BEGIN fields; }
<pos>" r "                      { printf("adve,["); BEGIN fields; }
<fields>(synset)                { printf("%s",yytext); BEGIN syn; }
<fields>.                       ;
<syn>"  "\n                     { printf("]).\n"); BEGIN 0; }
<syn>(synset)                   { printf(",%s",yytext); }
```

```
<syn>" "                    ;
(palabra)                   { printf("word('%s',",yytext); BEGIN pos; }

%%
```

A.3 Léxico del sistema

Se presenta a continuación el conjunto de hechos Prolog que forman el léxico utilizado en Ares, obtenido a partir de la base de datos de WordNet mediante los preprocesadores detallados en el anterior punto.

```
/* lexico de descripciones y de
   consultas */

word(['about'],adje,[00155141]).
word(['about'],adve,[00019458,0001
   9360,00004330]).
word(['access'],noun,[02750694,016
   37511]).
word(['access'],verb,[00897204,008
   09187]).
word(['accord'],noun,[06232515,035
   09380,03343660,02582438]).
word(['accord'],verb,[01066472,009
   00348]).
word(['account'],noun,[05962020,05
   961102,05941089,03526317,0333130
   9,03309943,03240793,03238974]).
word(['account'],verb,[00904682]).
word(['accounting'],noun,[05962020
   ,05961102,02974696,00219964]).
word(['activity'],noun,[06242704,0
   2552560,00140517]).
word(['add'],verb,[00928104,005413
   80,00411901,00376836,00246326,00
   212604,00073989]).
word(['adjacent'],adje,[00273234,0
   0217177]).
word(['administer'],verb,[00962615
   ,00922236,00035306]).
word(['administration'],noun,[0396
   2855,00386521,00237828]).
word(['advance'],adje,[00396097,00
   115410]).
word(['advance'],noun,[06131942,05
   946493,03580790,03574387,0350484
   3,00101359]).
word(['advance'],verb,[00948498,00
   911827,00803595,00803475,0080298
   9,00440447,00209916,00101998]).
word(['advanced'],adje,[01116968,0
   0937639,00918761,00763539,006083
   20,00397918]).
word(['affirmative'],adje,[0133750
   2,00044990,00492970]).
word(['affirmative'],noun,[0351947
   7]).
word(['alarm'],noun,[03639308,0335
   5896,01649059,01648368]).
word(['alarm'],verb,[00712224,0034
   3899]).
word(['alias'],noun,[03184083]).
word(['all'],adje,[01116032]).
word(['allow'],verb,[00958585,0090
   0470,00319663,00312395,00281882,
   00281463]).
word(['alter'],verb,[00671535,0008
   1649,00050777,00049698,00027518,
   00027158]).
word(['analysis'],noun,[03195965,0
   3100627,03090759,03012330,002382
   91,00228911]).
word(['another'],adje,[01034661,00
   777484]).
word(['application'],noun,[0323814
   9,02624882,00320406,00242015,002
   25461]).
word(['arbitrary'],adje,[00350021,
   00349328]).
word(['architecture'],noun,[031209
   09,01663179,00217919]).
word(['archive'],noun,[01664058]).
word(['archive'],verb,[00566122]).
word(['archives'],noun,[03239785,0
   1664058]).
word(['are'],noun,[06039303]).
word(['argument'],noun,[03512041,0
   3511568,03295998,03222792,030412
   09]).
word(['arithmetic'],noun,[03085745
   ]).
word(['arrive'],verb,[01025613,008
   08665]).
word(['as'],adve,[00053614,0000683
   6]).
word(['as'],noun,[06417852]).
word(['ascii'],noun,[03189552]).
word(['ask'],verb,[00304095,002908
   92]).
word(['assembler'],noun,[03260321
   ]).
word(['at'],noun,[06417928,0607147
   2]).
word(['attribute'],noun,[03038714,
   00015383]).
word(['attribute'],verb,[00282886]
   ).
word(['automatically'],adve,[00003
   492]).
word(['available'],adje,[00109237,
   00829643,00527931,00380178]).
word(['base'],adje,[01152473,00799
   416,00798081,00705151,00617236,0
   0441780]).
word(['base'],noun,[06411585,06211
   584,06027685,03016753,02190699,0
   2010872,01700265,01699933,016997
   56]).
word(['base'],verb,[00482358,00433
   311,00244941]).
word(['based'],adje,[01154175]).
word(['basic'],adje,[01128707,0092
   6198,00925506,00925222,00548047,
   00439424,00021878]).
word(['basic'],noun,[03406179]).
word(['batch'],noun,[06112472,0404
   4266,04044149]).
word(['batch'],verb,[00566712]).
word(['be'],noun,[06418240]).
word(['be'],verb,[01065635,0105319
   7,01049226,01035676,01034070,010
   33290,01032528,01030440,01029675
   ,00967974]).
word(['become'],verb,[01081204,010
   37421,01036766,00062187]).
word(['between'],adve,[00151797]).
word(['binary'],adje,[01268803,010
   92519]).
word(['binary'],noun,[04336933]).
word(['bit'],adje,[00659289]).
word(['bit'],noun,[06111175,060462
   43,04297522,03665703,03560920,02
   061348,01734384]).
word(['bite'],noun,[06318478,06318
   188,03665703,03664869,02678541,0
   0284911]).
word(['bite'],verb,[00847136,00591
   304,00591020]).
word(['block'],noun,[06220424,0413
   7051,03903885,02966471,02304474,
   02042375,01967540,01737603,01737
   063,01736948,01674558,00197489]
   .
word(['block'],verb,[01014997,0101
   4462,00625523,00602779,00602280,
   00601825,00458695,00424938,00415
   302,00201724,00201643]).
word(['browse'],noun,[03271876,002
   83945]).
word(['browse'],verb,[00928524,006
   43707,00536160,00469871]).
word(['buffer'],noun,[06476287,019
   83852,01768658,01768578,01768442
   ]).
word(['buffer'],verb,[00643814,002
   16815]).
word(['build'],noun,[02754975,0268
   1415]).
word(['build'],verb,[00667242,0019
   0027,00104163]).
word(['built'],adje,[01070721,0022
   5892]).
word(['byte'],noun,[06046502]).
word(['c'],adje,[01086084]).
word(['c'],noun,[06418961,06105640
   ,06085306,06049499,03405966,0185
   5924]).
word(['calculate'],verb,[00904682,
   00367325,00277110,00276567,00245
   059]).
word(['calculator'],noun,[04502425
   ,01784989]).
word(['calendar'],noun,[06634582,0
   4000940,03227926,03057904]).
word(['calendar'],verb,[00262674]
   ).
word(['call'],noun,[04040336,03515
   495,03515278,03514638,03489312,0
   3353826,03161223,03140647,003577
   36,00033454]).
word(['call'],verb,[00985787,00961
   866,00918667,00414071,00412619,0
   0386780,00362861,00307523,003065
   08,00009068]).
word(['cancel'],verb,[01075301,009
   81077,00980789,00633190,00311669
   ]).
word(['capability'],noun,[06387001
   ,02959644,02753013]).
word(['cat'],noun,[01814683,011094
   99,01103941]).
word(['cat'],verb,[00577086,000334
   76]).
word(['categorize'],verb,[00253237
   ,00170917]).
word(['change'],noun,[06190586,061
   63449,05952105,05951765,05951556
   ,03555614,01825047,00072928]).
word(['change'],verb,[00901469,000
   68165,00066589,00050777,00049698
   ,00046130]).
word(['character'],noun,[06358725,
   04510019,04403821,03368047,03315
   504,03064213,03038827,02547742]
   .
word(['character'],verb,[00538938]
   ).
word(['characteristic'],adje,[0018
   3514]).
word(['characteristic'],noun,[0336
   1930,03039127,02588211]).
word(['chart'],noun,[03445254,0188
   7255]).
word(['chart'],verb,[00276032,0027
   3850]).
word(['check'],noun,[06244919,0614
   0219,05949217,03244642,03241608,
   03030183,02997577,02574750,00427
   098,00227794,00197695]).
word(['check'],verb,[01013417,0100
   3435,00994472,00934677,00863927,
   00682701,00455355,00450786,00442
   014,00255573,00255471,00255279,0
   0187161,00150005,00142847,000962
   44]).
word(['clear'],adje,[00208396,0119
   9544,00952244,00814154,00703087,
   00669722,00661697,00608492,00462
   887,00436698,00422584,00381623,0
   0342030,00222331,00208193,001358
   50]).
word(['clear'],verb,[01087190,0096
```

```
7211,00915053,00914564,00841126,
00508598,00358550,00153439,00079
069,00073543]].
word(['click'],noun,[03583164,0348
8071,02244999]).
word(['click'],verb,[00873220,0087
3136,00759151,00500154,00423896,
00423657,00225187]).
word(['clock'],noun,[01845852]).
word(['clock'],verb,[00206957]).
word(['cluster'],noun,[03904728]).
word(['cluster'],verb,[00818222,00
605296]).
word(['code'],noun,[03303195,03189
102,03038222]).
word(['code'],verb,[00396075]).
word(['collate'],verb,[00566509,00
255206]).
word(['color'],noun,[03907767,0299
0197,02699877,02675985,02665437,
02569562]).
word(['color'],verb,[00684148,0068
3748,00119418]).
word(['column'],noun,[06213061,040
65124,04062027,03160581,01863722
,01863544,01863317]).
word(['combine'],noun,[03989392,03
581067,01864418]).
word(['combine'],verb,[01039555,00
921797,00596836,00566311,0016766
4,00078367]).
word(['command'],noun,[06139920,03
506089,03266508,02965262,0258481
3]).
word(['command'],verb,[01064586,00
966130,00406981,00290765,0029060
2]).
word(['commentary'],noun,[03340763
]).
word(['common'],adje,[00236473,002
33939,01263566,00837632,00799416
,00519130,00235121]).
word(['common'],noun,[04126324]).
word(['communication'],noun,[03152
810,00015070]).
word(['compact'],adje,[00240761,00
723396,00487176,00264486]).
word(['compact'],noun,[03343374,01
868222,01868081]).
word(['compact'],verb,[00605185,00
568261,00567732]).
word(['compare'],adve,[00108802]).
word(['compare'],verb,[00251928,00
251604]).
word(['comparison'],noun,[06176194
,00231484]).
word(['compiler'],noun,[04566857,0
3260587]).
word(['compress'],noun,[01692241]
).
word(['compress'],verb,[00568261,0
0567732]).
word(['concatenate'],verb,[0007685
0]).
word(['conditional'],adje,[0026751
0,00955702]).
word(['connect'],verb,[01036370,00
945723,00581509,00553185,0027733
6,00194374]).
word(['console'],noun,[01782519]).
word(['console'],verb,[00727308]).
word(['construct'],verb,[00667242,
00667051,00243380]).
word(['contain'],verb,[01066621,01
041369,01038828,00994472,0072094
9,00450786]).
word(['containing'],noun,[06353079
,06352951]).
word(['content'],adje,[00285527]).
word(['content'],noun,[06348320,06
113902,03904211,03644690,0327207
1,03023118,02332642]).
word(['content'],verb,[00728383,00
479228]).
word(['contents'],noun,[03228374]
).
word(['context'],noun,[06224787,04
103888,03165658]).
word(['contrast'],noun,[06188262,0
6176655,02988704,00231596]).
word(['contrast'],verb,[01054383,0
0255017]).
word(['control'],noun,[06360178,06
138044,04389171,03019060,0296526
2,02748221,02635502,01879239,002
72064]).
word(['control'],verb,[00999027,00
994472,00971631,00966130,0072185
9,00720949,00493233,00255573,002
27834,00096831]).
word(['conversion'],noun,[06163394
,03590915,03480122,00138056,0007
1793,00038501]).
word(['convert'],noun,[04531024]).
word(['convert'],verb,[00297975,00
163031,00161865,00066589]).
word(['copy'],noun,[03306831,03235
152,03202355,01888275]).
word(['copy'],verb,[01057886,01057
573,00700238,00699030,00683328]
).
word(['core'],noun,[04350539,04094
330,03990920,03274518,03061396,0
1891785,01891597]).
word(['core'],verb,[00648782]).
word(['correspond'],verb,[01050461
,00401812]).
word(['corresponding'],adje,[01030
519,00984839,00231908]).
word(['count'],noun,[06025874,0453
3763,00232086]).
word(['count'],verb,[01045874,0037
7468,00376240,00376051]).
word(['coverage'],noun,[05933885,0
3310893]).
word(['create'],verb,[00654089]).
word(['cross'],adje,[00722122,0056
8273]).
word(['cross'],noun,[03580964,0257
1193,02095235,01901511,00288547]
).
word(['cross'],verb,[01061333,0101
4849,00817537,00769111,00768900,
00768227]).
word(['current'],adje,[00323044,00
866587,00480817,00057702]).
word(['current'],noun,[04843989,03
587616]).
word(['currently'],adve,[00038424]
).
word(['cursor'],noun,[01911027]).
word(['curve'],noun,[06198283,0172
7088,00140005]).
word(['curve'],verb,[00822058,0082
1981,00821540,00154169]).
word(['data'],noun,[04070814]).
word(['database'],noun,[03292665])
.
word(['date'],noun,[06628440,06628
250,06609246,04542111,04040438,0
3784256]).
word(['date'],verb,[00984881,00984
762,00237332]).
word(['datum'],noun,[03026463]).
word(['decimal'],noun,[06098790,06
023442]).
word(['decode'],verb,[00396254,002
40487]).
word(['deep'],adje,[00935903,00337
667,00336286,01180636,00755681,0
0611619,00597914,00216370,001970
29]).
word(['deep'],noun,[04352351]).
word(['default'],noun,[05912536,00
026326]).
word(['default'],verb,[00899571]).
word(['delete'],verb,[00633190,003
98344,00070485]).
word(['delta'],noun,[04319425,0337
2313]).
word(['deny'],verb,[00885053,00319
085,00318917]).
word(['dependency'],noun,[06256542
,06241343,06145563]).
word(['desk'],noun,[01925017]).
word(['determine'],verb,[00375643,
00365235,00295760,00272213,00271
932,00271678,00270820]).
word(['device'],noun,[02093964,016
09642,00063703]).
word(['dictionary'],noun,[03212641
]).
word(['difference'],noun,[06185594
,03577010,03511443,03043487]).
word(['different'],adje,[01031260,
01034661,00822747]).
word(['different'],adve,[00015885]
).
word(['directory'],noun,[03214387]
).
word(['disable'],verb,[00216430,00
039728]).
word(['disk'],noun,[06201721,02253
580,02163913,01935609]).
word(['disk'],verb,[00698763]).
word(['diskette'],noun,[01935772])
.
word(['display'],noun,[03523640,03
399499,01936514,01816714,0017901
5]).
word(['display'],verb,[01004577,00
855169]).
word(['distinguish'],verb,[0087661
2,00255017,00251349,00251156,002
50917]).
word(['distribution'],noun,[030962
45,02717819,00378677,00367393]).
word(['do'],noun,[03605098]).
word(['do'],verb,[01071505,0105576
9,01034508,01018225,01015998,010
15621,00738915,00691099,00664342
,00654930,00018689]).
word(['document'],noun,[05958854,0
3223459,01939390]).
word(['document'],verb,[00399580,0
0257557]).
word(['doing'],noun,[00039771]).
word(['domain'],noun,[06382039,041
00611,04099739,03906526]).
word(['draw'],noun,[06183832,04556
073,04322102,01946825,01946683,0
0196356,00170246,00046150]).
word(['draw'],verb,[00923077,00916
154,00803879,00744062,00724147,0
0681814,00681436,00663628,006452
44,00628883,00594054,00592067,00
500839,00481682,00467600,0044232
6,00429476,00393549,00285177]).
word(['drive'],noun,[02617911,0195
1048,01950972,01950857,00269995,
00203400,00200797,00109383,00042
101]).
word(['drive'],verb,[00951669,0083
0980,00776944,00776729,00618401]
).
word(['driver'],noun,[04557002,045
56774,01951108]).
word(['dump'],noun,[05999693,04101
134]).
word(['dump'],verb,[00936116,00889
057]).
word(['duplicate'],adje,[00742145]
).
word(['duplicate'],noun,[06184525,
01888504]).
word(['duplicate'],verb,[00697138,
00160765]).
word(['dynamic'],adje,[01352569,00
394119]).
word(['each'],adje,[01116436]).
word(['each'],adve,[00143698]).
word(['early'],adje,[00398598,0039
5732,00865396,00826834,00822902]
).
word(['early'],adve,[00040835,0004
0557]).
word(['echo'],noun,[03518084,02687
060]).
word(['echo'],verb,[01057675,00872
494,00380789]).
word(['edit'],verb,[00241644,00081
863,00081315]).
word(['editing'],noun,[03215544,00
430090]).
word(['editor'],noun,[04561687,032
60933]).
word(['affective'],adje,[00407410,
01221455,00913349,00887542,00408
068,00406925,00338058,00246133]
).
word(['eject'],verb,[00839487,0059
9294,00562903,00045283]).
word(['electronic'],adje,[01273734
,01273657]).
word(['eliminate'],verb,[01038625,
00199681,00199429,00032676,00032
504]).
word(['enable'],verb,[00216670]).
word(['encode'],verb,[00396075]).
word(['encrypt'],verb,[00396075]).
word(['end'],adje,[00793580,005021
80]).
word(['end'],noun,[06683975,061670
67,04567084,04103698,04103604,04
103307,04103192,03566682,0307923
5,01967086,01628428,00246697]).
word(['end'],verb,[01061009,010319
38,00150111,00148650]).
word(['entry'],noun,[05960713,0350
6026,03234902,01942212,00018848]
).
word(['environment'],noun,[0638181
3,04103888]).
word(['error'],noun,[03342501,0305
1853,02621904,02605050,00029738,
00029287]).
word(['establish'],verb,[00960542,
```

```
00665113,00664963,00256936,00256
660,00244941]).
word(['evaluate'],verb,[00264149])
.
word(['ex'],noun,[04615365,0455245
5]).
word(['examine'],verb,[01003157,00
857656,00306134,00305220,0024829
0]).
word(['exchange'],noun,[06163449,0
6158108,03494208,01972494,018206
47,00396036,00371205,00078043]).
word(['exchange'],verb,[00901469,0
0066589,00058275]).
word(['execute'],verb,[01016285,01
015621,00984018,00983489,0069109
9,00662251,00397674]).
word(['execution'],noun,[06017115,
00394523,00383098,00039771]).
word(['exerciser'],noun,[02048644]
).
word(['exit'],noun,[03567621,01973
507,00022422]).
word(['exit'],verb,[00813660,00429
770,00152125]).
word(['expand'],verb,[00839670,001
30676,00109631,00097041]).
word(['expanded'],adje,[00267385,0
0266864]).
word(['expression'],noun,[03499620
,03496445,03466959,03396749,0332
8332,02569233]).
word(['extend'],verb,[01062444,010
62176,01061556,01061333,00918147
,00589089,00563780,00465600,0013
5463,00127790,00099257,00097041,
00013137,00012988]).
word(['extract'],noun,[03278680,00
473843]).
word(['extract'],verb,[00744536,00
658972,00551805,00094391]).
word(['facility'],noun,[02965743,0
2965362,02580839,02482593,015761
03]).
word(['factor'],noun,[06127546,060
96527,06025937,03566241]).
word(['factor'],verb,[00246239]).
word(['false'],adje,[01202687,0120
5405,01082112,00790368,00585788,
00558162,00475649,00307237]).
word(['feature'],noun,[03279900,03
039127,02953915]).
word(['feature'],verb,[01039849,01
039069]).
word(['few'],adje,[00772962]).
word(['field'],noun,[06382039,0485
0595,04318724,04105476,04105121,
04104963,03920242,03918541,03918
428,03084235,03065908,00372825])
.
word(['field'],verb,[00430787,0043
0536,00318452]).
word(['file'],noun,[04061749,03236
400,01988053,01987896]).
word(['file'],verb,[00771303,00567
289,00399396,00399307,00342636])
.
word(['filter'],noun,[01989834,019
89281]).
word(['filter'],verb,[00836682,005
95574,00595439]).
word(['find'],noun,[03022354,00058
066]).
word(['find'],verb,[00912731,00897
075,00885166,00860379,00851067,0
0846030,00661158,00387377,003652
35,00280389,00278114,00277969,00
221202]).
word(['finish'],noun,[06683975,035
72873,02576845,00078791]).
word(['finish'],verb,[01031938,005
12464,00480579,00204173,00149760
,00149246,00148650]).
word(['first'],adje,[00499895,0108
7771,00520280]).
word(['first'],adve,[00154008,0004
2524,00041813]).
word(['first'],noun,[06683526,0617
0346,06027409,01993884,00245041]
).
word(['float'],noun,[06696309,0592
9164,03893035,01615418,00041954]
).
word(['float'],verb,[00764155,0076
3537,00752157,00269890]).
word(['floating'],adje,[01154637,0
1153986,01065070,00595649,000962
17,00046305]).
word(['flow'],noun,[06692676,06010
439,05998734,03587252,00116653])
.
word(['flow'],verb,[00835087,00834
840]).
word(['fold'],noun,[06216424,03914
374,02899684,02007770,02007679,0
0140291]).
word(['fold'],verb,[00960299,00618
734,00518970]).
word(['font'],noun,[03369176,01695
081]).
word(['foreign'],adje,[00514904,00
512760,00674987]).
word(['form'],noun,[03989924,03949
357,03606853,03223961,03167153,0
3064829,03036166,02754975,027105
88,02568333,00012039]).
word(['form'],verb,[01036766,01035
676,00968883,00962177,00668801,0
0666433,00058811]).
word(['format'],noun,[03292498,025
68222]).
word(['format'],verb,[00699705]).
word(['frame'],noun,[06679645,0294
4679,02754975,02398470,02012934,
02012312,02012030]).
word(['frame'],verb,[01072866,0102
2230,00647334,00391170,00274580]
).
word(['free'],adje,[00529546,00526
843,00525880,01233107,00851714,0
0815716,00815613,00814154,004470
86,00292193,00140175,00096307]).
word(['free'],verb,[01016401,00981
929,00958188,00957767,00935999,0
0924743,00923563,00624244,006016
36,00357775]).
word(['front'],adje,[00115156,0008
0048]).
word(['front'],noun,[06147690,0485
2870,04587087,04137352,04107580,
04107425,03068886,01977458]).
word(['front'],verb,[01063597,0042
9861]).
word(['function'],noun,[06114511,0
4044750,03264710,02740331,002441
55]).
word(['function'],verb,[00942730,0
0622982]).
word(['general'],adje,[00714794,00
550369,00547426,00237210]).
word(['general'],noun,[04590068]).
word(['generate'],verb,[00658110,0
0657907,00657780,00025481]).
word(['generation'],noun,[06676683
,04034165,04034002,00288437]).
word(['generator'],noun,[04590468,
02030973]).
word(['get'],verb,[00920518,008843
79,00843513,00808665,00663628,00
586061,00461197,00298581,0024385
5,00221202,00062187,00048957,000
48806,00040905,00037764]).
word(['give'],noun,[02691525]).
word(['give'],verb,[00931735,00924
978,00922076,00921574,00892362,0
0890999,00879807,00879619,008789
53,00741296,00696536,00665113,00
658303,00658110,00593082,0047165
8,00426843,00426671,00426539,002
85740,00260884,00184762]).
word(['given'],adje,[00558947,0064
6143,00546539,00022631]).
word(['given'],adve,[00125847]).
word(['given'],noun,[03050585]).
word(['graph'],noun,[03445254]).
word(['graphics'],noun,[03444684,0
2337012]).
word(['grind'],noun,[04776950,0022
1118,00125872]).
word(['grind'],verb,[00956993,0082
8170,00651022,00570910,00141046]
).
word(['group'],adje,[01058978,0023
6767]).
word(['group'],noun,[06413087,0309
2751,00014600]).
word(['group'],verb,[00433687,0025
2906]).
word(['hard'],adje,[00581381,00579
224,00577305,00365738,01176996,0
1142432,01112025,00598018,005818
98,00564245,00532119]).
word(['hard'],adve,[00097831,00036
295,00036144,00036013,00035913,0
0035817,00035677]).
word(['has'],verb,[01039069,009411
89,00881984,00881868,00881223,00
843513,00696536,00583321,0046162
3,00048957,00029911,00029688,000
26231]).
word(['have'],noun,[04731459]).
word(['have'],verb,[01039069,00941
189,00881984,00881868,00881223,0
0843513,00696536,00583321,004616
23,00048957,00029911,00029688,00
026231]).
word(['help'],noun,[06177462,04472
766,02740498,00410178]).
word(['help'],verb,[01010698,00473
213,00035577]).
word(['history'],noun,[06609760,03
238974,03128938,03128807]).
word(['host'],noun,[04655861,04613
002,04612820,03967493,03732023,0
0466970]).
word(['host'],verb,[00479332]).
word(['how'],adve,[00047971,000478
59]).
word(['icon'],noun,[02258030]).
word(['id'],noun,[06064576,0339929
8,02955449]).
word(['idle'],adje,[00155783,01154
984]).
word(['idle'],verb,[00956375,00623
245]).
word(['image'],noun,[03482416,0306
7519,03063402,02568787,02258030,
01887777]).
word(['image'],verb,[00851408,0066
0505]).
word(['immune'],adje,[01278189,011
59364]).
word(['immune'],noun,[04617414]).
word(['implement'],noun,[01615966]
).
word(['implement'],verb,[01015299,
00952369]).
word(['improve'],verb,[00084165,00
083667]).
word(['improved'],adje,[00643103])
.
word(['in'],adje,[00471607,0090798
0]).
word(['in'],adve,[00106119]).
word(['in'],noun,[06422396]).
word(['incoming'],adje,[00647377,0
0867440,00548876]).
word(['incoming'],noun,[00018848])
.
word(['indent'],verb,[00952570,005
19926,00519813,00518588]).
word(['index'],noun,[03360924,0329
4023,03228544,02936394]).
word(['indicate'],verb,[00368284,0
0368116,00366206,00365598,002991
45]).
word(['information'],noun,[0353296
8,03291565,03026197]).
word(['initialize'],verb,[00272101
1]).
word(['input'],noun,[03544488,0303
1968,00114621]).
word(['inquiry'],noun,[03515898,03
018306,00226344]).
word(['insert'],verb,[00581779,005
68403,00410769,00075520]).
word(['inspect'],verb,[00865043,00
270400]).
word(['install'],verb,[00943552,00
641326]).
word(['interactive'],adje,[0030230
4,00973112]).
word(['interface'],noun,[02099054,
02098926]).
word(['interpolate'],verb,[0024727
3,00081649]).
word(['interpretation'],noun,[0350
7257,03063696,03005711,00040594]
).
word(['interpreter'],noun,[0476413
6,04622395,03261263]).
word(['interval'],noun,[06684647,0
3383940,03092174,02716918]).
word(['invert'],verb,[00163406,001
63275]).
word(['inverted'],adje,[01033577])
.
word(['invoke'],verb,[00658822,003
08043]).
word(['is'],verb,[01065635,0105319
7,01049226,01035676,01034070,010
33290,01032528,01030440,01029675
,00967974]).
word(['item'],adve,[00155163]).
word(['item'],noun,[03027563,01573
621]).
word(['job'],noun,[06345703,035795
```

```
59,03033791,02109333,02109166,00
264919,00246745,00208030]).
word(['job'],verb,[01020148,009741
70,00957313,00907324]).
word(['join'],noun,[06218491,03918
026]).
word(['join'],verb,[01036370,00963
587,00526573,00525225]).
word(['key'],adje,[00637488,004070
99]).
word(['key'],noun,[06091343,043255
39,03387882,03031720,02533138,02
115288,02115152]).
word(['key'],verb,[00931327,002513
49]).
word(['keyboard'],noun,[02115521,0
2115411]).
word(['lack'],noun,[06364114,02725
544]).
word(['lack'],verb,[01040001,00476
752]).
word(['lacking'],adje,[00923462,00
032924]).
word(['language'],noun,[03164973,0
3022552,02969189]).
word(['large'],adje,[00696979,0069
1740,00766564,00733637,00554188]
).
word(['last'],adje,[00501429,00866
071,00520462]).
word(['last'],adve,[00022467,00022
378]).
word(['last'],noun,[06683975,06088
013,06042252]).
word(['last'],verb,[01068836,01034
765]).
word(['leave'],noun,[06618820,0623
9130,03313871,00360926,00020216]
).
word(['leave'],verb,[01079047,0104
1940,00943253,00890328,00810387,
00431167,00235274,00235075,00209
564,00152572,00056069]).
word(['letter'],noun,[03370596,033
19310,03285514]).
word(['letter'],verb,[00914948,006
82445,00682345]).
word(['level'],adje,[00619729,0044
4742,00444314]).
word(['level'],noun,[06354148,0622
4046,06155037,02719510,02136926,
02000244,01999854]).
word(['level'],verb,[00669513,0053
2825,00460646,00460556,00151422]
).
word(['lexical'],adje,[01314578,01
314422]).
word(['library'],noun,[03912112,02
138474]).
word(['like'],adje,[01029335,01028
589,00433776]).
word(['like'],adve,[00053614]).
word(['like'],verb,[00734420,00734
341,00732392,00709329]).
word(['line'],noun,[06215444,06196
031,06176744,06084403,05947983,0
4119027,04118401,04070000,040627
66,03943867,03454984,03449961,03
154671,03287065,03008798,0246542
7,02293709,02222118,02143869,021
43672,02142516,01875377,01866672
,00409306,00207077]).
word(['line'],verb,[01067931,00645
244,00518214,00515547,00193035,0
0091662]).
word(['link'],noun,[06199026,06058
531,06055720,03156420,02314173,0
2145391,02145261]).
word(['link'],verb,[01036370,00608
974,00553185,00277336]).
word(['links'],noun,[01578523]).
word(['list'],noun,[05965543,03227
258]).
word(['list'],verb,[00978599,00892
478,00375313]).
word(['listing'],noun,[03227258,00
339717]).
word(['load'],noun,[02520852,01801
016,01617026]).
word(['load'],verb,[00609678,00608
509,00608387,00607212]).
word(['local'],adje,[01309318,0130
9218,00714639,00552811,00551648,
00514454]).
word(['local'],noun,[02153221,0162
0825]).
word(['locate'],verb,[01064401,009
12938,00175377,00860097]).
word(['location'],noun,[02715142,0
2714847,00355812,00060056,000117
95]).
word(['log'],adje,[01100084]).
word(['log'],noun,[06601787,033612
69,03234360,02156074]).
word(['log'],verb,[00509075,003997
29]).
word(['logical'],adje,[00716418,01
221717,00971626,00224076]).
word(['long'],adje,[00719081,00717
620,01140856,00214979,00062687]
).
word(['long'],adve,[00034640]).
word(['long'],verb,[00734246,00723
002]).
word(['look'],noun,[02569233,02567
119,00297534]).
word(['look'],verb,[00860097,00852
988,00851624,00016034]).
word(['lookup'],noun,[06014311]).
word(['low'],adje,[00611387,006092
64,00606403,01148235,01123307,00
945599,00799416,00798269,0072742
3,00612517,00441780]).
word(['low'],noun,[06384213,027199
10,01993884]).
word(['low'],verb,[00423804]).
word(['machine'],noun,[04648078,03
999886,02162185,02161086,0179683
4]).
word(['macro'],adje,[00693919]).
word(['macro'],noun,[03266728]).
word(['magnetic'],adje,[01330724,0
0101050]).
word(['magnify'],verb,[00328500,00
184659,00098963]).
word(['mail'],noun,[03284631,03157
430,02165570,01822621]).
word(['mail'],verb,[00588024,00414
179]).
word(['maintain'],verb,[01059940,0
0910414,00881765,00474736,004067
23,00406345,00354778,00286016]).
word(['make'],noun,[00119498]).
word(['make'],verb,[01035676,01015
621,00948021,00915053,00914564,0
0816371,00816085,00696536,006683
32,00664662,00664342,00661991,00
655790,00654930,00654089,0029858
1,00285177,00116044,00048806,000
32828,00032250]).
word(['manager'],noun,[04520093,04
457822]).
word(['manual'],adje,[01280194,001
09022,01096237,00889806]).
word(['manual'],noun,[03213827]).
word(['map'],noun,[06115390,018869
34]).
word(['map'],verb,[00680501,002760
32,00161657]).
word(['mark'],noun,[06074580,04513
724,03547513,03365098,03354542,0
3352584,03031504,02997399,025749
56,02571193,00024977]).
word(['mark'],verb,[00993777,00993
677,00846109,00648339,00634052,0
0518095,00400509,00365761,003118
62,00255471,00253459,00251156,00
234863,00214979,00076747]).
word(['massage'],noun,[00234412]).
word(['massage'],verb,[00495627,00
029090]).
word(['master'],adje,[01150102,010
95290]).
word(['master'],noun,[04796992,046
85449,04655579,04655480,04655384
,04655255,04648569,04604917,0217
7223]).
word(['master'],verb,[01006504,004
39534,00227951,00227834]).
word(['mathematics'],noun,[0308550
2]).
word(['media'],noun,[03156648]).
word(['medium'],adje,[00761367,002
99542]).
word(['medium'],noun,[04658357,041
21485,03153598,02179263]).
word(['membership'],noun,[04044459
]).
word(['memory'],noun,[03066146,030
03670,02969625,02183853]).
word(['merge'],verb,[00167664,0015
6546,00156425]).
word(['merging'],adje,[00297053]).
word(['merging'],noun,[06148537,00
133253]).
word(['message'],noun,[03272071,03
153425]).
word(['meter'],noun,[06060543,0347
4552,02677621,02187102]).
word(['meter'],verb,[00516745,0020
6647]).
word(['mode'],noun,[06123325,03094
717,02656174]).
word(['modification'],noun,[061912
38,06122989,03555614,00074357]).
word(['modified'],adje,[00764934,0
0955825,00181968]).
word(['modify'],verb,[00069218,000
68437]).
word(['more'],adje,[00777017,00775
037]).
word(['more'],adve,[00040368,00040
154,00015885]).
word(['more'],noun,[06022158]).
word(['motion'],noun,[06242539,048
34446,03561064,03504274,03395486
,00117874,00100582]).
word(['motion'],verb,[00395347]).
word(['mouse'],noun,[02205055,0128
3192]).
word(['mouse'],verb,[00768094,0048
6889]).
word(['move'],noun,[00062765]).
word(['move'],verb,[00950309,00744
658,00742092,00737009,00734868,0
0706197,00705287,00428731,003468
12,00050155,00006761]).
word(['mp'],noun,[04661601]).
word(['much'],adje,[00773885,00598
413,00303360]).
word(['much'],adve,[00018574,00010
395,00007173]).
word(['much'],noun,[06108356]).
word(['multi'],adje,[00772691]).
word(['multiple'],adje,[01091977]
).
word(['multiple'],noun,[03041737]
).
word(['name'],noun,[06358820,04670
384,03908872,03325003,03182386]
).
word(['name'],verb,[00948021,00947
893,00892478,00412619,00411201,0
0410056,00386780,00375643,002513
49,00248896]).
word(['network'],noun,[04065459,02
213916,02186142]).
word(['network'],verb,[00414675]).
word(['new'],adje,[00822126,005313
63,00399785,00399607]).
word(['newly'],adve,[00050465,0005
0382,00050303]).
word(['ni'],noun,[06424547]).
word(['nice'],adje,[00796277,00993
516,00918124,00485978,00378694,0
0101995,00052019]).
word(['notice'],noun,[03512955,033
51637,03333377,02988358,02986056
]).
word(['notice'],verb,[00860379,008
46109,00425856,00425460]).
word(['numb'],adje,[01052973]).
word(['numb'],verb,[00844684]).
word(['number'],noun,[06024996,034
01261,03357906,03271417,03214913
,03214647,02727942]).
word(['number'],verb,[01045514,003
76569,00376240,00097444]).
word(['numbering'],noun,[03229168]
).
word(['numeric'],adje,[01281472,00
956858]).
word(['object'],noun,[03172564,030
80054,03024528,00007642]).
word(['object'],verb,[00314786]).
word(['obtain'],verb,[01047052,008
93565,00221202]).
word(['off'],adje,[01160760,008281
49]).
word(['off'],adve,[00110446,001103
72]).
word(['on'],adje,[00827789]).
word(['on'],adve,[00023486]).
word(['one'],adje,[01212219,010914
06,01090994,01083704,01030855]).
word(['one'],noun,[06101202]).
word(['operator'],noun,[06115557,0
4680905,04680680,04615474]).
word(['option'],noun,[05886422,030
16041,00032928]).
word(['optional'],adje,[00412034,0
0883151]).
word(['order'],noun,[06378282,0623
1406,04035445,04035280,03985154,
03947181,03506089,03297683,03251
243,03243743,02718695,02642592,0
0339171]).
```

```
word(['order'],verb,[00994818,00944357,00288662,00288461,00288069,00253555,00116466]).
word(['ordering'],noun,[06169948,00339171]).
word(['other'],adje,[01034459,00866071,00777753]).
word(['other'],adve,[00015885]).
word(['out'],adje,[01027228,00471801,00923391,00908926,00881996,00276485,00058700]).
word(['out'],adve,[00140137,00106025,00105933]).
word(['out'],noun,[00051157]).
word(['output'],noun,[03544593,02293945,00309452]).
word(['over'],adje,[00865281,00763240,00251624,00250862]).
word(['over'],adve,[00135523,00135410,00135324]).
word(['own'],adje,[00886722]).
word(['own'],verb,[00881868]).
word(['owner'],noun,[04685566,04611113]).
word(['ownership'],noun,[05885691,00273198]).
word(['page'],noun,[03154415]).
word(['page'],verb,[00954148,00291732]).
word(['pair'],noun,[06101565,03912395,03912012]).
word(['pair'],verb,[00986456,00584037,00525859]).
word(['panel'],noun,[04052882,04019432,03497845,02237613,02237299,01880304]).
word(['paper'],noun,[06557134,03208997,03159076,03153766]).
word(['paper'],verb,[00513907,00513812]).
word(['parameter'],noun,[03041358]).
word(['parse'],verb,[00248034]).
word(['part'],adje,[00250387]).
word(['part'],adve,[00028594]).
word(['part'],noun,[06126769,05906572,04296727,04132577,03456105,03064213,02978634,02768637,02757486,01628198,00244155]).
word(['part'],verb,[00962443,00820365,00812770,00637142,00636744]).
word(['parts'],noun,[04126727]).
word(['password'],noun,[03305829]).
word(['pattern'],noun,[06167939,03068005,03064829,03054293,02976682,01627885,00141573]).
word(['pattern'],verb,[01073192,00699285]).
word(['pc'],noun,[02251945]).
word(['perform'],verb,[00939855,00691616,00691099]).
word(['performance'],noun,[03579376,03400945,03005711,00192709,00039771]).
word(['perhaps'],adve,[00051807]).
word(['permission'],noun,[06139050,03313658,00388084]).
word(['permit'],noun,[06139050,03313984,01496572]).
word(['permit'],verb,[00312395]).
word(['permute'],verb,[00161500]).
word(['physical'],adje,[01301241,00889154,01056599,00886289,00790731,00304907]).
word(['piece'],noun,[05906768,04296727,03715044,03560920,03459205,03159908,02369832,02258747,02044212,01828046,01573798]).
word(['plot'],noun,[04150616,03194475,03057033]).
word(['plot'],verb,[00683115,00275165]).
word(['plotter'],noun,[04739432,04704618,04530339]).
word(['point'],noun,[06639512,06224046,06219248,06214031,06083912,06036581,04328548,04129184,03380347,03365954,03276200,03044818,03027563,02713628,02588396,02275758]).
word(['point'],verb,[01073591,00777274,00775540,00460556,00460361,00289706,00166590,00108277]).
word(['portable'],adje,[00760115,00645355]).
word(['portion'],noun,[06128052,05906572,03665239,01628198,00368041]}).
word(['portion'],verb,[00921881,00890194]).
word(['postscript'],noun,[03285925,03204677]).
word(['precision'],noun,[02605489]).
word(['prepare'],verb,[00945045,00670656,00666374,00210566,00230444,00171940,00087090]).
word(['prevent'],verb,[00970659,00969947]).
word(['previous'],adje,[00865710,00078348]).
word(['print'],noun,[03307585,02335909,01594409]).
word(['print'],verb,[00700997,00700530,00700026,00699939]).
word(['printer'],noun,[04713682,02291438]).
word(['printing'],noun,[03307328,03269181,03205971,00375123]).
word(['priority'],noun,[06353588,06166300]).
word(['procedure'],noun,[03264710,00343846,00343517]).
word(['process'],noun,[05974150,03257330,02985424,02888752,00343846]).
word(['process'],verb,[01023928,00964733,00804351,00672118,00217774]).
word(['processing'],noun,[06012802]).
word(['profile'],noun,[04124968,04096576,03446567,03240057]).
word(['program'],noun,[03333965,03282290,03259414,03232942,03053144]).
word(['program'],verb,[00700409]).
word(['protocol'],noun,[03038222]).
word(['provide'],verb,[00940322,00928901,00473834]).
word(['provided'],adje,[00545835]).
word(['public'],adje,[00929673,00236950]).
word(['public'],noun,[03965863,03906406]).
word(['purpose'],noun,[04290219,03079488,02740331]).
word(['purpose'],verb,[00275437,00271575]).
word(['put'],noun,[00033605]).
word(['put'],verb,[00610308,00609940,00391170]).
word(['query'],noun,[03515898]).
word(['query'],verb,[00304377]).
word(['queue'],noun,[04063977,03233731,02770136]).
word(['queue'],verb,[00822893]).
word(['quit'],verb,[01059741,00943253,00889636,00809740,00431167]).
word(['quota'],noun,[06140739,06028378,05908734]).
word(['random'],adje,[00544173,00378180,00350556,00261385]).
word(['raster'],noun,[01817424]).
word(['read'],verb,[00365947,00240962,00240821,00240560,00239629,00239270,00229327,00226654]).
word(['rearrange'],verb,[00116892]).
word(['receive'],verb,[00987768,00884097,00843323,00599859,00357127,00221202]).
word(['recovery'],noun,[06000667,03594177,00037943]).
word(['reference'],noun,[06119486,03341435,03341063,03315504,03305040,03274358,03212286,03061708]).
word(['reference'],verb,[00687650]).
word(['regard'],noun,[06358492,03652208,03626473,03027759,00298280]).
word(['regard'],verb,[01058577,00852822]).
word(['regenerate'],adje,[00977781]).
word(['regenerate'],verb,[00659093,00068016,00067901,00042073]).
word(['regular'],adje,[00982490,00978537,01131119,00543890,00530494,00235312]).
word(['regular'],noun,[04728152]).
word(['relate'],verb,[01081325,00378849,00277336]).
word(['related'],adje,[00983950,00982899,00986147,00842744,00007260]).
word(['relation'],noun,[06130102,04418761,03527378,00408412,00015659]).
word(['remind'],verb,[00346311,00233977]).
word(['reminder'],noun,[04458193,03235729,03025346]).
word(['remote'],adje,[00707080,00216639,00214526,00214352,00015482]).
word(['remove'],verb,[00983048,00951139,00951020,00179903,00070485]).
word(['rename'],verb,[00066589,00050777,00049698]).
word(['replicate'],verb,[00697138]).
word(['reply'],noun,[03517858,03333159]).
word(['reply'],verb,[00318569]).
word(['report'],noun,[03525423,03525298,03309943,03208997]).
word(['report'],verb,[00385333,00384369,00384259,00384116,00383981]).
word(['representation'],noun,[06178965,03331589,03062865,01885654,00413286,00304551]).
word(['request'],noun,[03512750,03238700]).
word(['request'],verb,[00291345,00290892]).
word(['reset'],noun,[02338422]).
word(['reset'],verb,[00125547]).
word(['resolve'],verb,[00876769,00408682,00271575,00271299,00270672]).
word(['respect'],noun,[06358492,03626278,03027759,02650946,00417946,00385648]).
word(['respect'],verb,[00972350,00269104]).
word(['restore'],verb,[01013022,00960453,00922735,00107939,00068016]).
word(['retrieve'],verb,[00896948,00586479,00232770]).
word(['return'],noun,[06175409,05959061,05894658,03604692,03517973,00421357,00199828,00196543,00116319,00097248,00036443,00019633]).
word(['return'],verb,[00949892,00922388,00912392,00808438,00807983,00658110,00586376,00585945,00381535,00318797,00163513]).
word(['reverse'],adje,[00117026]).
word(['reverse'],noun,[06191418,06189786,02340571,00123360,00027202]).
word(['reverse'],verb,[00311329,00278509,00163275,00056780]).
word(['ripple'],noun,[03570725]).
word(['ripple'],verb,[00874066,00824355]).
word(['root'],adje,[00926198]).
word(['root'],noun,[06095206,05768722,04078821,03920606,03167539,02794613]).
word(['root'],verb,[00520836,00152018]).
word(['rout'],noun,[03967577,00027110]).
word(['rout'],verb,[00520836,00520514,00439807]).
word(['route'],noun,[04127031,02343568]).
word(['run'],noun,[06175676,04357323,03608456,03602596,03588008,00268879,00196161,00109217,00109113,00105204,00072004]).
word(['run'],verb,[01078949,01078517,01077891,01061556,01046853,01046747,01032770,01016091,01000916,00967127,00958089,00895018,00838663,00835087,00832579,00775307,00775216,00775000,00774709,00769229,00747515,
00662457,00623128,00622982]).
word(['running'],adje,[01112376,00544541,00288604]).
word(['running'],noun,[00386767,00150357,00105204]).
word(['s'],noun,[06428098,06152184
```

```
,06080442,06049853]).
word(['same'],adje,[01033871,01033
753,01030236,00602792,00433776,0
0182797]).
word(['scan'],verb,[01052029,00859
474,00859350,00744385,00421996,0
0241174,00240962]).
word(['screen'],noun,[02395369,023
93501,02367453,02366653,02366548
,02241557,01816892]).
word(['screen'],verb,[01003435,009
49660,00855062,00602056,00596549
,00450129,00270540]).
word(['script'],noun,[03448992,032
06082,03189912]).
word(['search'],noun,[06014311,030
07508,00318430,00231967,00226037
]).
word(['search'],verb,[00860097,005
37001,00536270,00250111]).
word(['secret'],adje,[01045426,009
28658,00849785,00438500,00203439
,00023030]).
word(['secret'],noun,[03305467]).
word(['see'],adve,[00108802]).
word(['see'],noun,[04097662]).
word(['see'],verb,[00987432,009848
81,00851408,00851195,00851067,00
817330,00660505,00267999,0025557
3,00228671,00225097]).
word(['segment'],noun,[06127117,02
376610]).
word(['segment'],verb,[00639338]).
word(['select'],adje,[01151547,011
49623]).
word(['select'],verb,[00261605]).
word(['selection'],noun,[06011066,
04043798,03278680,03015566,00061
345]).
word(['send'],verb,[00935066,00785
620,00785403,00587901,00433311,0
0414179]).
word(['sender'],noun,[04745353]).
word(['server'],noun,[04799971,047
47231,02378250]).
word(['service'],noun,[03974641,03
969433,02740498,02378342,0041110
2,00346952,00289275,00208299,002
05322,00205179,00200907,00112676
,00096509,00040311]).
word(['service'],verb,[00583708,00
255279,00194267]).
word(['session'],noun,[06662862,04
054401,03497915]).
word(['set'],adje,[01218506,009645
31,00953668,00760728,00578038,00
363760,00295580]).
word(['set'],noun,[06680154,041400
72,03990186,03918646,03917217,03
141047,02979279,02425498,0232301
9,00222191]).
word(['set'],verb,[00792197,006907
80,00688016,00640300,00625014,00
610308,00503461,00442231,0037564
3,00271678,00187690,00172650,001
71940,00125657,00125380,00124655
,00025849,00018689]).
word(['setting'],adje,[01216287]).
word(['setting'],noun,[06381945,04
138780,04103888,02204777,0219645
5]).
word(['share'],noun,[06128194,0593
2794,05906572,02274048,00368041]
).
word(['share'],verb,[01075926,0091
7183,00917059,00916791]).
word(['shared'],adje,[01073838,010
59642,00237357]).
word(['shell'],noun,[06460915,0498
5630,02385145,02365016,02384850,
02268361,00918093]).
word(['shell'],verb,[00453346,0008
0161]).
word(['show'],noun,[03573238,03453
024,03400392,03399703,03282290,0
1936514,00178642]).
word(['show'],verb,[00857486,00857
134,00854808,00853952,00806540,0
0405901,00374372,00366206,003659
47,00256660]).
word(['side'],adje,[00926857]).
word(['side'],noun,[06211662,04359
754,04140410,04140297,04080264,0
4048947,03715392,02926971,019773
37]).
word(['side'],verb,[01069691]).
word(['signal'],adje,[00641554]).
word(['signal'],noun,[03350617]).
word(['signal'],verb,[00416790,003
65598]).
word(['simple'],adje,[01078596,000
67127,00925222,00924628,00919189
,00896274,00837632,00413665,0036
8639,00341147,00090254,00086640]
).
word(['single'],adje,[01090832,011
08537,01079090,00739095,00238670
]).
word(['single'],noun,[06101202,000
52963]).
word(['single'],verb,[00575789]).
word(['size'],adje,[01093939]).
word(['size'],noun,[06438377,02720
554]).
word(['size'],verb,[00252610,00135
942]).
word(['smooth'],adje,[01103654,010
97687,01127625,00682916,00569989
,00444878,00368370,00311665,0015
9187,00135850,00119490]).
word(['smooth'],verb,[00506288,005
02327]).
word(['soft'],adje,[00727127,00582
173,00581574,00580639,00578356,0
1196733,00882982,00598596,003682
62]).
word(['sort'],noun,[06014169,03036
166]).
word(['sort'],verb,[00949660,00252
348]).
word(['sorted'],adje,[01094066,002
02406]).
word(['source'],noun,[06167696,046
19504,04590468,04079004,04078821
,03565997,03306405,02413328]).
word(['space'],adje,[01180050]).
word(['space'],noun,[06218224,0414
2154,03919146,03377039,03202066,
00012745]).
word(['space'],verb,[00800982]).
word(['specified'],adje,[00546350]
).
word(['specify'],verb,[01032416,00
409080,00407536,00375643,0027823
5,00275666,00262368]).
word(['spell'],noun,[06694896,0634
4949,03501948]).
word(['spell'],verb,[00685086,0037
1851,00369653]).
word(['spelling'],noun,[03188559]
).
word(['split'],adje,[01214442,0121
3187,01181558,00659898,00152777]
).
word(['split'],noun,[06218982,0014
9609,00149301,00136621,00134549]
).
word(['split'],verb,[00976728,0096
2443,00820365,00636651,00130428]
).
word(['spool'],noun,[01739962]).
word(['spool'],verb,[00621303]).
word(['standard'],adje,[01128062,0
0799805,00574810,00234381]).
word(['standard'],noun,[06173637,0
6042164,05945334,03542242,030620
53,02092768]).
word(['start'],noun,[06683526,0617
7750,04142910,03602473,03565626,
03351143,02376776,00293622,00087
534]).
word(['start'],verb,[01031725,0094
1733,00812770,00767245,00767143,
00747614,00745586,00665945,00657
993,00147604,00146969]).
word(['state'],adje,[00930947,0093
0820]).
word(['state'],noun,[04143031,0409
4755,03965511,03964439,00012937]
).
word(['state'],verb,[00403100,0034
6396]).
word(['status'],noun,[06226287,061
70127,06160053,02607001]).
word(['stick'],noun,[06143865,0579
4020,02434039,02279198,02109963]
).
word(['stick'],verb,[00624101,0055
4068,00490954,00289144]).
word(['stop'],noun,[06244919,04144
506,03576810,03487188,01928545,0
1815673,01737603,00365349,003577
98,00084855]).
word(['stop'],verb,[01059741,01014
997,00747042,00746625,00746044,0
0589219,00450786,00154273,001486
50]).
word(['store'],noun,[05942974,0238
9510,02183853,02006201,01694480,
00129589]).
word(['store'],verb,[00911010,0091
0638,00609479]).
word(['stream'],noun,[06174978,043
51218,03587616,00116653]).
word(['stream'],verb,[01082789,008
36590,00836477,00819764]).
word(['string'],noun,[06174696,034
50292,02440981,01891440,01890788
,01750340,01709685,01709563]).
word(['string'],verb,[00803374,005
56200,00556081,00555960,00555738
,00555504,00076961]).
word(['strip'],noun,[04297123,0344
7012,02442598,02017630,01647922,
00182030]).
word(['strip'],verb,[00933515,0092
4226,00634656,00511241,00511049,
00205825,00079069,00078969,00078
835,00078693,00078481,00023399]
).
word(['stuck'],adje,[00097384,0088
5852]).
word(['subsequent'],adje,[00077157
]).
word(['summary'],adje,[00820501,00
264486]).
word(['summary'],noun,[03222063]).
word(['sun'],noun,[06631097,048442
95,04338848]).
word(['super'],adje,[01149973]).
word(['super'],noun,[04774132]).
word(['suspend'],verb,[00991684,00
603630,00154758,00061615]).
word(['switch'],adje,[01012049]).
word(['switch'],noun,[03603264,024
53466,02452610,02452445,02050048
,00198314,00074082]).
word(['switch'],verb,[00902078,006
15639,00161343,00058275,00049333
]).
word(['symbol'],noun,[03357484,030
69399]).
word(['symbolic'],adje,[01309103,0
1287105,00711179]).
word(['syntax'],noun,[06170856,031
35425,03135283]).
word(['system'],noun,[06560988,040
65789,03054786,02995100,02974379
,02760709,02755816,01633090]).
word(['tab'],noun,[03241608]).
word(['table'],noun,[04318514,0400
0431,03658795,02455867]).
word(['table'],verb,[01044936]).
word(['tag'],noun,[03548819,023162
46,00166919,00056706]).
word(['tag'],verb,[00806917,006860
13,00653538,00648339,00413226]).
word(['take'],noun,[05894658,00307
188]).
word(['take'],verb,[01038123,00947
066,00906004,00892589,00883983,0
0883522,00883211,00882983,008822
22,00806226,00739651,00647110,00
586063,00487910,00461623,0029266
6,00277788,00261605,00259203,002
39270,00229327,00222138,00070485
]).
word(['talk'],noun,[03534979,03527
750,03494555,00303129]).
word(['talk'],verb,[00383296,00382
848,00378702,00375338]).
word(['tape'],noun,[02461756,02461
314,02461136]).
word(['tape'],verb,[00542695,00400
095,00397952]).
word(['temporarily'],adve,[0004487
9,00034123]).
word(['terminal'],adje,[01288196,0
1288125,00502180,00491017]).
word(['terminal'],noun,[02472152,0
2471610,02199794]).
word(['terminate'],verb,[01061009,
01031938,00950835,00150111,00148
650]).
word(['termination'],noun,[0007831
9]).
word(['test'],adje,[00418353]).
word(['test'],noun,[06173940,03517
171,03019359,03007756,02997577,0
0918093,00338058,00268879]).
word(['test'],verb,[01003435,01003
157,00305103]).
word(['text'],noun,[03210873,03201
596]).
word(['through'],adje,[00374521,00
251088]).
word(['time'],noun,[06684977,06675
```

```
222,06617340,06614763,06610265,0
3560779,03551862,02677621,000124
18]).
word(['time'],verb,[00206957]).
word(['times'],noun,[06609371]).
word(['trace'],noun,[06111992,0350
4711,03031395,02488296,02335703]
).
word(['trace'],verb,[00807690,0080
7412,00645244,00284061]).
word(['traffic'],noun,[06157658,04
059435]).
word(['traffic'],verb,[00903051]).
word(['transfer'],noun,[05973173,0
3001913,00377656,00112544,001121
77,00075382]).
word(['transfer'],verb,[00946797,0
0891522,00888205,00887964,008118
34,00744300,00587377]).
word(['translate'],verb,[00587585,
00381779,00226654,00220963,00162
672]).
word(['translation'],noun,[0360378
4,03216264,03187945,00138056,001
17608]).
word(['trivial'],adje,[01061786,00
837526,00639595,00304142]).
word(['true'],adje,[01200960,00704
667,00585377,00556816,00474638,0
0306559,00020234]).
word(['true'],adve,[00102729]).
word(['truth'],noun,[03326266,0302
7148,02628385]).
word(['turn'],noun,[06694896,06680
238,06198746,03596169,03589559,0
1401261,00416879,00156533,001228
70,00120850,00104944,00074357]).
word(['turn'],verb,[01037421,00766
868,00765768,00646033,00587795,0
0297975,00195012,00118713,000596
72,00056780,00050551]).
word(['two'],adje,[01083807]).
word(['two'],noun,[06101565]).
word(['type'],noun,[03949538,03917
669,03369109,03353398,03036166]
).
word(['type'],verb,[00400757]).
word(['typeset'],verb,[00699538]).
word(['unauthorized'],adje,[001074
38,00820771]).
word(['under'],adje,[00607062,0038
7917]).
word(['underline'],noun,[03355410]
).
word(['underline'],verb,[00405685]
).
word(['undo'],verb,[00655076,00619
910,00066476]).
word(['unix'],noun,[03259253]).
word(['unload'],verb,[00607972,006
07073]).
word(['up'],adje,[01217164,0121656
7,01215648,00544624,00251624,001
10702]).
word(['up'],adve,[00039152]).
word(['up'],verb,[00063672]).
word(['update'],noun,[03295276]).
word(['update'],verb,[00326280,000
69026,00068894]).

word(['usage'],noun,[00319314,0014
2490]).
word(['use'],noun,[06003284,027404
98,02740331,00385990,00319314,00
142996]).
word(['use'],verb,[01019061,004930
91,00465393,00463027,00462789]).
word(['used'],adje,[00013444,00821
766,00014059]).
word(['user'],noun,[04792616,04571
964]).
word(['using'],noun,[00146404]).
word(['utility'],adje,[01153243,00
927537]).
word(['utility'],noun,[03969169,02
740091]).
word(['validate'],verb,[00981374,0
0404813,00257955,00190409]).
word(['value'],noun,[06616487,0304
0694,02990667,02733982]).
word(['value'],verb,[00900912,0026
9748,00269104,00264149,00254829,
00208360]).
word(['variable'],adje,[01224490,0
0179260]).
word(['variable'],noun,[03040912]
).
word(['various'],adje,[01225888,01
032390]).
word(['version'],noun,[03507745,03
187945,03063696]).
word(['vi'],adje,[01084039]).
word(['vi'],noun,[06102374]).
word(['vice'],noun,[02623868,00253
754]).
word(['visible'],adje,[01229909,00
849305,00109417]).
word(['visual'],adje,[01307842,013
04710,01230491]).
word(['wait'],noun,[06687882,00359
588]).
word(['wait'],verb,[01044559,01043
136,00954475,00279686]).
word(['warn'],verb,[00343021,00342
777]).
word(['warning'],adje,[00887846]).
word(['warning'],noun,[03527880,03
305172]).
word(['when'],adve,[00100915]).
word(['width'],noun,[02732655]).
word(['window'],noun,[02534293,025
34046,02533777,01817257]).
word(['with'],adve,[00119594]).
word(['within'],adve,[00048664]).
word(['word'],noun,[06046679,03528
630,03506368,03496588,03331185,0
3305829,03295056,03166153]).
word(['word'],verb,[00390865]).
word(['words'],noun,[03512264,0348
4700,03465058,03449458]).
word(['work'],noun,[04849198,03001
234,02295468,02295370,01575620,0
0208030,00204747]).
word(['work'],verb,[01001052,00967
974,00954587,00953059,00952081,0
0828708,00698530,00672118,006229
82,00497332,00043456]).
word(['working'],adje,[00419930,00
025152]).

word(['write'],verb,[00699459,0068
5086,00684811,00682035,00414179,
00401940,00395646]).
word(['written'],adje,[01262213]).
word(['x'],adje,[01084318]).
word(['x'],noun,[06103736]).
word(['yet'],adve,[00009554,000094
75,00009371,00089292]).

% vocabulario incluido de forma
  semiautomática:

% keyword := key + word
word(['keyword'],noun,[06091343,04
325539,03387882,03031720,0253313
8,02115288,02115152,06046679,035
28630,03506368,03496588,03331185
,03305829,03295056,03166153]).
% pathname := path+name
word([pathname],noun,[04309450,041
27031,02243684,00143372,06358820
,04670384,03908872,03325003,0318
2386]).

% overwrite := write + remove
word(['overwrite'],verb,[00699459,
00685086,00684811,00682035,00414
179,00401940,00395646,00983048,0
0951139,00951020,00179903,000704
85]).

% look for := seek, search
word(['look','for'],verb,[00536270
]).


% palabras no incluidas en
  WordNet:

word([W],noun,[W]) :-
  word_def(L),
  member(W,L).
word_def(WL) :-
  WL =
  [automatically,beautifier,been,b
  ibliographic,checksum,coff,color
  map,cpp,crontab,csh,currently,db
  x,debugger,decrypt,desktop,deskt
  ops,diplay,disassembler,ed,emula
  tor,encryption,eqn,ethernet,exec
  utables,expander,filenames,forma
  tters,hangups,has,hexadecimal,ho
  stname,how,ibm,identifier,if,ifd
  ef,internet,interprocess,ip,iso,
  keystroke,lineprinter,logged,log
  in,logins,newly,nroff,numlock,oc
  tal,overstrike,per,perhaps,photo
  typesetter,preprocessor,printabl
  e,processor,programmable,rasterf
  ile,relational,relocation,repeti
  tively,rpc,scanning,sccs,semapho
  re,sunview,tbl,tcp,tektronix,tel
  net,temporarily,terminfo,textedi
  t,tfs,that,them,they,topological
  ,troff,typescript,uncompress,upo
  n,username,uucp,verifier,versa,v
  ersatec,vfont,what,which,whiteou
  t,who,within,yacc,yet,you].
```

A.4 Elementos de datos de entrada

Los siguientes son los hechos Prolog con el conjunto de descripciones de órdenes de la sección 1 del manual del S.O. y el conjunto de consultas utilizado en los experimentos.

A.4.1 Descripciones de órdenes

```
command(name('acctcom'),
  description(['search','and','pri
  nt','process','accounting','file
  s']) ).
command(name('adb'),
  description(['general','purpose'
  ,'debugger']) ).
command(name('addbib'),
  description(['create','or','exte
  nd','a','bibliographic','databas
  e']) ).
command(name('adjacentscreens'),
  description(['connect','multiple
  ','screens','to','sunview','wind
  ow','driver']) ).
command(name('admin'),
  description(['create','and','adm
  inister','sccs','history','files
  ']) ).
command(name('aedplot'),
  description[['graphics','filters
  ','for','various','plotters']]
  ).
command(name('align'),
```

```
  description(['filters','provided
  ','with','textedit']) ).
command(name('apropos'),
  description(['locate','commands'
  ,'by','keyword','lookup']) ).
command(name('ar'),
  description(['create','library',
  'archives',',','and','add','or',
  'extract','files']) ).
command(name('arch'),
  description(['display','the','ar
  chitecture','of','the','current'
  ,'host']) ).
command(name('as'),
  description(['assembler']) ).
command(name('at'),
  description(['execute','a','comm
  and','or','script','at','a','spe
  cified','time']) ).
command(name('atq'),
  description(['display','the','qu
  eue','of','jobs','to','be','run'
  ,'at','specified','times']) ).
command(name('atrm'),
  description(['remove','jobs','sp
  ooled','by','at','or','batch'])
  ).
command(name('awk'),
  description(['pattern','scanning
  ','and','processing','language']
  ) ).
command(name('banner'),
  description(['display','a','stri
  ng','in','large','letters']) ).
command(name('bar'),
  description(['create','tape','ar
  chives',',','and','add','or','ex
  tract','files']) ).
command(name('basename'),
  description(['display','portions
  ','of','pathnames','and','filena
  mes']) ).
command(name('batch'),
  description(['execute','a','comm
  and','or','script','at','a','spe
  cified','time']) ).
command(name('bc'),
  description(['arbitrary','precis
  ion','arithmetic','language'])
  ).
command(name('bgplot'),
  description(['graphics','filters
  ','for','various','plotters'])
  ).
command(name('biff'),
  description(['give','notice','of
  ','incoming','mail','messages'])
  ).
command(name('bin'),
  description(['an','early','progr
  am','for','processing','mail','m
  essages']) ).
command(name('cal'),
  description(['display','a','cale
  ndar']) ).
command(name('calendar'),
  description(['a','simple','remin
  der','service']) ).
command(name('cancel'),
  description(['send','cancel','re
  quests','to','a','printer']) ).
command(name('capitalize'),
  description(['filters','provided
  ','with','textedit']) ).
command(name('cat'),
  description(['concatenate','and'
  ,'display']) ).
command(name('cb'),
  description(['a','simple','c','p
  rogram','beautifier']) ).
command(name('cc'),
  description(['c','compiler']) ).
command(name('cd'),
  description(['change','working',
  'directory']) ).
command(name('cdc'),
  description(['change','the','del
  ta','commentary','of','an','sccs
  ','delta']) ).
command(name('cflow'),
  description(['generate','a','flo
  w','graph','for','a','c','progra
  m']) ).
command(name('checkeq'),
  description(['typeset','mathemat
  ics']) ).
command(name('checknr'),
  description(['check','nroff','an
  d','troff','input','files','for'
  ,'errors']) ).
command(name('chfn'),
  description(['change','local','o
  r','nis','password','information
  ']) ).
command(name('chgrp'),
  description(['change','the','gro
  up','ownership','of','a','file']
  ) ).
command(name('chkey'),
  description(['create','or','chan
  ge','encryption','key']) ).
command(name('chmod'),
  description(['change','the','per
  missions','mode','of','a','file'
  ]) ).
command(name('chsh'),
  description(['change','local','o
  r','nis','password','information
  ']) ).
command(name('clear'),
  description(['clear','the','term
  inal','screen']) ).
command(name('clear_colormap'),
  description(['clear','the','colo
  rmap','to','make','console','tex
  t','visible']) ).
command(name('clear_functions'),
  description(['reset','the','sele
  ction','service','to','clear','s
  tuck','function','keys']) ).
command(name('click'),
  description(['enable','or','disa
  ble','the','keyboard','s','keyst
  roke','click']) ).
command(name('clock'),
  description(['display','the','ti
  me','in','an','icon','or','windo
  w']) ).
command(name('cluster'),
  description(['find','optional','
  cluster','containing','a','file'
  ]) ).
command(name('cmdtool'),
  description(['run','a','shell','
  or','program','using','the','sun
  view','text','facility']) ).
command(name('cmp'),
  description(['perform','a','byte
  ','by','byte','comparison','of',
  'two','files']) ).
command(name('col'),
  description(['filter','reverse',
  'paper','motions','for','termina
  l','display']) ).
command(name('colcrt'),
  description(['filter','nroff','o
  utput','for','a','terminal','lac
  king','overstrike','capability']
  ) ).
command(name('coloredit'),
  description(['alter','color','ma
  p','segment']) ).
command(name('colrm'),
  description(['remove','character
  s','from','specified','columns',
  'within','each','line']) ).
command(name('comb'),
  description(['combine','sccs','d
  eltas']) ).
command(name('comm'),
  description(['display','lines','
  in','common','between','two','so
  rted','lists']) ).
command(name('compress'),
  description(['compress','or','ex
  pand','files',',','display','exp
  anded','contents']) ).
command(name('cp'),
  description(['copy','files']) ).
command(name('cpio'),
  description(['copy','file','arch
  ives','in','and','out']) ).
command(name('cpp'),
  description(['the','c','language
  ','preprocessor']) ).
command(name('crontab'),
  description(['install',',','edit
  ',',','remove','or','list','a','
  user','s','crontab','file']) ).
command(name('crtplot'),
  description(['graphics','filters
  ','for','various','plotters'])
  ).
command(name('crypt'),
  description(['encode','or','deco
  de','a','file']) ).
command(name('csh'),
  description(['shell','with','a',
  'c','like','syntax','and','advan
  ced','interactive','features'])
  ).
command(name('csplit'),
  description(['split','a','file',
  'with','respect','to','a','given
  ','context']) ).
command(name('ctags'),
  description(['create','a','tags'
  ,'file','for','use','with','ex',
  'and','vi']) ).
command(name('ctrace'),
  description(['generate','a','c',
  'program','execution','trace'])
  ).
command(name('cu'),
  description(['connect','to','rem
  ote','system']) ).
command(name('cut'),
  description(['remove','selected'
  ,'fields','from','each','line','
  of','a','file']) ).
command(name('cxref'),
  description(['generate','a','c',
  'program','cross','reference'])
  ).
command(name('date'),
  description(['display','or','set
  ','the','date']) ).
command(name('dbx'),
  description(['source','level','d
  ebugger']) ).
command(name('dbxtool'),
  description(['sunview','interfac
  e','for','dbx','source','level',
  'debugger']) ).
command(name('dc'),
  description(['desk','calculator'
  ]) ).
command(name('dd'),
  description(['convert','and','co
  py','files','with','various','da
  ta','formats']) ).
command(name('defaultsedit'),
  description(['edit','default','s
  ettings','for','sunview','utilit
  ies']) ).
command(name('defaults_from_input'
  ),
  description(['update','current',
  'state','of','the','mouse','and'
  ,'keyboard']) ).
command(name('defaults_to_mailrc')
  ,
  description(['edit','default','s
  ettings','for','sunview','utilit
  ies']) ).
command(name('delta'),
  description(['make','a','delta',
  'to','an','sccs','file']) ).
command(name('deroff'),
  description(['remove','nroff',',
  ','troff',',',',','tbl','and','eqn',
  'constructs']) ).
command(name('des'),
  description(['encrypt','or','dec
  rypt','data','using','data','enc
  ryption','standard']) ).
command(name('desktop'),
  description(['switch','the','win
  dow','system','to','be','invoked
  ','upon','login']) ).
command(name('df'),
  description(['report','free','di
  sk','space','on','file','systems
  ']) ).
command(name('diff'),
  description(['display','line','b
  y','line','differences','between
  ','pairs','of','text','files'])
  ).
command(name('diffmk'),
  description(['mark','differences
  ','between','versions','of','a',
  'troff','input','file']) ).
command(name('dircmp'),
  description(['compare','director
  ies']) ).
command(name('dirname'),
  description(['display','portions
  ','of','pathnames','and','filena
  mes']) ).
command(name('dis'),
  description(['object','code','di
  sassembler','for','coff']) ).
command(name('disablenumlock'),
  description(['enable','or','disa
  ble','the','numlock','key']) ).
command(name('domainname'),
```

```
  description(['set','or','display
  ','name','of','the','current','n
  is','domain']) ).
command(name('dos'),
  description(['sunview','window',
  'for','ibm','pc','at','applicati
  ons']) ).
command(name('dos2unix'),
  description(['convert','text','f
  ile','from','dos','format','to',
  'iso','format']) ).
command(name('du'),
  description(['display','the','nu
  mber','of','disk','blocks','used
  ','per','directory','or','file']
  ) ).
command(name('dumbplot'),
  description(['graphics','filters
  ','for','various','plotters'])
  ).
command(name('dumpkeys'),
  description(['load','and','dump'
  ,'keyboard','translation','table
  s']) ).
command(name('e'),
  description(['line','editor'])
  ).
command(name('echo'),
  description(['echo','arguments',
  'to','the','standard','output'])
  ).
command(name('ed'),
  description(['basic','line','edi
  tor']) ).
command(name('edit'),
  description(['line','editor'])
  ).
command(name('egrep'),
  description(['search','a','file'
  ,'for','a','string','or','regula
  r','expression']) ).
command(name('eject'),
  description(['eject','media','de
  vice','from','drive']) ).
command(name('enablenumlock'),
  description(['enable','or','disa
  ble','the','numlock','key']) ).
command(name('enroll'),
  description(['send','or','receiv
  e','secret','mail']) ).
command(name('env'),
  description(['obtain','or','alte
  r','environment','variables'])
  ).
command(name('eqn'),
  description(['typeset','mathemat
  ics']) ).
command(name('error'),
  description(['categorize','compi
  ler','error','messages',',','ins
  ert','at','source','file','lines
  ']) ).
command(name('ex'),
  description(['line','editor'])
  ).
command(name('expand'),
  description(['expand','tab','cha
  racters','to','space','character
  s',',','and','vice','versa']) ).
command(name('expr'),
  description(['evaluate','express
  ions','as','logical',',',',arithm
  etic',',','or','string']) ).
command(name('false'),
  description(['provide','truth','
  values']) ).
command(name('fdformat'),
  description(['format','diskettes
  ']) ).
command(name('fgrep'),
  description(['search','a','file'
  ,'for','a','string','or','regula
  r','expression']) ).
command(name('file'),
  description(['determine','the','
  type','of','a','file','by','exam
  ining','its','contents']) ).
command(name('find'),
  description(['find','files','by'
  ,'name',',',',or','by','other','c
  haracteristics']) ).
command(name('finger'),
  description(['display','informat
  ion','about','users']) ).
command(name('fmt'),
  description(['simple','text','an
  d','mail','message','formatters'
  ]) ).
command(name('fmt_mail'),
  description(['simple','text','an
  d','mail','message','formatters'
  ]) ).
command(name('fold'),
  description(['fold','long','line
  s','for','display']) ).
command(name('fontedit'),
  description(['a','vfont','screen
  ','font','editor']) ).
command(name('foption'),
  description(['determine','availa
  ble','floating','point','code','
  generation','options']) ).
command(name('from'),
  description(['display','the','se
  nder','and','date','of','newly',
  'arrived','mail','messages']) ).
command(name('ftp'),
  description(['file','transfer','
  program']) ).
command(name('gcore'),
  description(['get','core','image
  s','of','running','processes'])
  ).
command(name('get'),
  description(['retrieve','a','ver
  sion','of','an','sccs','file'])
  ).
command(name('get_alarm'),
  description(['sunview','programm
  able','alarms']) ).
command(name('getoptcvt'),
  description(['parse','command','
  options','in','shell','scripts']
  ) ).
command(name('getopt'),
  description(['parse','command','
  options','in','shell','scripts']
  ) ).
command(name('getopts'),
  description(['parse','command','
  options','in','shell','scripts']
  ) ).
command(name('get'),
  description(['copy','contents','
  of','sunview','selection','to','
  the','standard','output']) ).
command(name('gfxtool'),
  description(['run','graphics','p
  rograms','in','a','sunview','win
  dow']) ).
command(name('gigiplot'),
  description(['graphics','filters
  ','for','various','plotters'])
  ).
command(name('gprof'),
  description(['display','call','g
  raph','profile','data']) ).
command(name('graph'),
  description(['draw','a','graph']
  ) ).
command(name('grep'),
  description(['search','a','file'
  ,'for','a','string','or','regula
  r','expression']) ).
command(name('groups'),
  description(['display','a','user
  ','s','group','memberships']) ).
command(name('hashcheck'),
  description(['report','spelling'
  ,'errors']) ).
command(name('hashmake'),
  description(['report','spelling'
  ,'errors']) ).
command(name('head'),
  description(['display','first','
  few','lines','of','specified','f
  iles']) ).
command(name('help'),
  description(['help','regarding',
  'sccs','error','or','warning','m
  essages']) ).
command(name('help'),
  description(['provide','help','w
  ith','sunview','applications','a
  nd','desktop']) ).
command(name('hostid'),
  description(['print','the','nume
  ric','identifier','of','the','cu
  rrent','host']) ).
command(name('hostname'),
  description(['set','or','print',
  'name','of','current','host','sy
  stem']) ).
command(name('hpplot'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('i'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('iapx'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('iconedit'),
  description(['create','and','edi
  t','images','for','icons',',','c
  ursors','and','panel','items'])
  ).
command(name('id'),
  description(['print','the','user
  ','name','and','id',',',',and','g
  roup','name','and','id']) ).
command(name('implot'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('indent'),
  description(['indent','and','for
  mat','a','c','program','source',
  'file']) ).
command(name('indentpro'),
  description(['edit','default','s
  ettings','for','sunview','utilit
  ies']) ).
command(name('indxbib'),
  description(['create','an','inve
  rted','index','to','a','bibliogr
  aphic','database']) ).
command(name('inline'),
  description(['in','line','proced
  ure','call','expander']) ).
command(name('input'),
  description(['edit','default','s
  ettings','for','sunview','utilit
  ies']) ).
command(name('input'),
  description(['update','the','cur
  rent','state','of','the','mouse'
  ,'and','keyboard']) ).
command(name('insert'),
  description(['filters','provided
  ','with','textedit']) ).
command(name('install'),
  description(['install','files'])
  ).
command(name('ipcrm'),
  description(['remove','a','messa
  ge','queue',',',',semaphore','set
  ',',',',or','shared','memory','id
  ']) ).
command(name('ipcs'),
  description(['report','interproc
  ess','communication','facilities
  ','status']) ).
command(name('join'),
  description(['relational','datab
  ase','operator']) ).
command(name('keylogin'),
  description(['decrypt','and','st
  ore','secret','key']) ).
command(name('keylogout'),
  description(['delete','stored','
  secret','key']) ).
command(name('kill'),
  description(['send','a','signal'
  ,'to','a','process',',',',or','te
  rminate','a','process']) ).
command(name('last'),
  description(['indicate','last','
  logins','by','user','or','termin
  al']) ).
command(name('lastcomm'),
  description(['show','the','last'
  ,'commands','executed',',',',in',
  'reverse','order']) ).
command(name('ld'),
  description(['link','editor',',',
  'dynamic','link','editor']) ).
command(name('ldd'),
  description(['list','dynamic','d
  ependencies']) ).
command(name('ld.so'),
  description(['link','editor',',',
  'dynamic','link','editor']) ).
command(name('leave'),
  description(['remind','you','whe
  n','you','have','to','leave'])
  ).
command(name('lex'),
  description(['lexical','analysis
  ','program','generator']) ).
command(name('line'),
  description(['read','one','line'
```

```
]) ).
command(name('lint'),
  description(['a','c','program','
  verifier']) ).
command(name('ln'),
  description(['make','hard','or',
  'symbolic','links','to','files']
  ) ).
command(name('load'),
  description(['load','clusters'])
  ).
command(name('loadc'),
  description(['load','clusters'])
  ).
command(name('loadkeys'),
  description(['load','and','dump'
  ,'keyboard','translation','table
  s']) ).
command(name('lockscreen'),
  description(['edit','default','s
  ettings','for','sunview','utilit
  ies']) ).
command(name('lockscreen'),
  description(['maintain','sunview
  ','context','and','prevent','una
  uthorized','access']) ).
command(name('lockscreen'),
  description(['maintain','sunview
  ','context','and','prevent','una
  uthorized','access']) ).
command(name('logger'),
  description(['add','entries','to
  ','the','system','log']) ).
command(name('login'),
  description(['log','in','to','th
  e','system']) ).
command(name('logname'),
  description(['get','the','name',
  'by','which','you','logged','in'
  ]) ).
command(name('look'),
  description(['find','words','in'
  ,'the','system','dictionary','or
  ','lines','in','a','sorted','lis
  t']) ).
command(name('lookbib'),
  description(['find','references'
  ,'in','a','bibliographic','datab
  ase']) ).
command(name('lorder'),
  description(['find','an','orderi
  ng','relation','for','an','objec
  t','library']) ).
command(name('lp'),
  description(['send','cancel','re
  quests','to','a','printer']) ).
command(name('lpq'),
  description(['display','the','qu
  eue','of','printer','jobs']) ).
command(name('lpr'),
  description(['send','a','job','t
  o','the','printer']) ).
command(name('lprm'),
  description(['remove','jobs','fr
  om','the','printer','queue']) ).
command(name('lpstat'),
  description(['display','the','pr
  inter','status','information'])
  ).
command(name('lptest'),
  description(['generate','linepri
  nter','ripple','pattern']) ).
command(name('ls'),
  description(['list','the','conte
  nts','of','a','directory']) ).
command(name('lsw'),
  description(['list','tfs','white
  out','entries']) ).
command(name('m'),
  description(['macro','language',
  'processor']) ).
command(name('m'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('mach'),
  description(['display','the','pr
  ocessor','type','of','the','curr
  ent','host']) ).
command(name('machid'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('mail'),
  description(['read','or','send',
  'mail','messages']) ).
command(name('mailrc'),
  description(['edit','default','s
  ettings','for','sunview','utilit
  ies']) ).
command(name('mailtool'),
  description(['sunview','interfac
  e','for','the','mail','program']
  ) ).
command(name('make'),
  description(['maintain',',',',upd
  ate',',','and','regenerate','rel
  ated','programs','and','files'])
  ).
command(name('man'),
  description(['display','referenc
  e','manual','pages']) ).
command(name('mesg'),
  description(['permit','or','deny
  ','messages','on','the','termina
  l']) ).
command(name('mkdir'),
  description(['make','a','directo
  ry']) ).
command(name('mkstr'),
  description(['create','an','erro
  r','message','file','by','massag
  ing','c','source','files']) ).
command(name('more'),
  description(['browse','or','page
  ','through','a','text','file'])
  ).
command(name('mps'),
  description(['display','status',
  'of','current','processes','on',
  'an','mp','system']) ).
command(name('mpstat'),
  description(['show','multi','pro
  cessor','usage']) ).
command(name('mt'),
  description(['magnetic','tape','
  control']) ).
command(name('mv'),
  description(['move','or','rename
  ','files']) ).
command(name('nawk'),
  description(['pattern','scanning
  ','and','processing','language']
  ) ).
command(name('neqn'),
  description(['typeset','mathemat
  ics']) ).
command(name('newgrp'),
  description(['log','in','to','a'
  ,'new','group']) ).
command(name('nice'),
  description(['run','a','command'
  ,'at','low','priority']) ).
command(name('nl'),
  description(['line','numbering',
  'filter']) ).
command(name('nm'),
  description(['print','symbol','n
  ame','list']) ).
command(name('nohup'),
  description(['run','a','command'
  ,'immune','to','hangups','and','
  quits']) ).
command(name('nroff'),
  description(['format','commands'
  ,'for','display','or','line','pr
  inter']) ).
command(name('objdump'),
  description(['dump','selected','
  parts','of','a','coff','object',
  'file']) ).
command(name('od'),
  description(['octal',',','decima
  l',',','hexadecimal',',','and','
  ascii','dump']) ).
command(name('old-ccat'),
  description(['compress','and','u
  ncompress','files',',','and','ca
  t','them']) ).
command(name('old-compact'),
  description(['compress','and','u
  ncompress','files',',','and','ca
  t','them']) ).
command(name('old-eyacc'),
  description(['modified','yacc','
  allowing','much','improved','err
  or','recovery']) ).
command(name('old-filemerge'),
  description(['window','based','f
  ile','comparison','and','merging
  ','program']) ).
command(name('old-make'),
  description(['maintain',',',',upd
  ate',',','and','regenerate','gro
  ups','of','programs']) ).
command(name('old-prmail'),
  description(['display','waiting'
  ,'mail']) ).
command(name('old-pti'),
  description(['phototypesetter','
  interpreter']) ).
command(name('old-setkeys'),
  description(['modify','interpret
  ation','of','the','keyboard'])
  ).
command(name('old-sun3cvt'),
  description(['convert','sun',',','
  ,'system','executables','to','sun
  ',',','system','executables'])
  ).
command(name('old-syslog'),
  description(['make','a','system'
  ,'log','entry']) ).
command(name('old-uncompact'),
  description(['compress','and','un
  compress','files',',',',','and','ca
  t','them']) ).
command(name('old-vc'),
  description(['version','control'
  ]) ).
command(name('on'),
  description(['execute','command'
  ,'on','a','remote','system','wit
  h','local','environment']) ).
command(name('organizer'),
  description(['file','and','direc
  tory','manager']) ).
command(name('overview'),
  description(['run','a','program'
  ,'from','sunview','that','takes'
  ,'over','the','screen']) ).
command(name('pack'),
  description(['compress','and','e
  xpand','files']) ).
command(name('page'),
  description(['browse','or','page
  ','through','a','text','file'])
  ).
command(name('pagesize'),
  description(['display','the','si
  ze','of','a','page','of','memory
  ']) ).
command(name('passwd'),
  description(['change','local','o
  r','nis','password','information
  ']) ).
command(name('paste'),
  description(['join','correspondi
  ng','lines','of','files',',',',su
  bsequent','lines','of','one'])
  ).
command(name('pax'),
  description(['portable','archive
  ','exchange']) ).
command(name('paxcpio'),
  description(['copy','file','arch
  ives','in','and','out']) ).
command(name('pcat'),
  description(['compress','and','e
  xpand','files']) ).
command(name('pdp'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('perfmeter'),
  description(['display','system',
  'performance','values','in','a',
  'meter','or','strip','chart'])
  ).
command(name('pg'),
  description(['page','through','a
  ','file','on','a','soft','copy',
  'terminal']) ).
command(name('plot'),
  description(['graphics','filters
  ','for','various','plotters'])
  ).
command(name('pr'),
  description(['prepare','file','f
  or','printing',',',',perhaps','in
  ','multiple','columns']) ).
command(name('printenv'),
  description(['display','environm
  ent','variables','currently','se
  t']) ).
command(name('prof'),
  description(['display','profile'
  ,'data']) ).
command(name('prs'),
  description(['display','selected
  ','portions','of','an','sccs','h
  istory']) ).
command(name('prt'),
  description(['display','delta','
```

```
table','information','from','an'
,'sccs','file']) ).
command(name('ps'),
  description(['display','the','st
atus','of','current','processes'
]) ).
command(name('ptx'),
  description(['generate','a','per
muted','index']) ).
command(name('pwd'),
  description(['display','the','pa
thname','of','the','current','wo
rking','directory']) ).
command(name('quota'),
  description(['display','a','user
','s','disk','quota','and','usag
e']) ).
command(name('ranlib'),
  description(['convert','archives
','to','random','libraries']) ).
command(name('rasfilter'),
  description(['convert','an',',',
'bit','deep','rasterfile','to','
a',',','bit','deep','rasterfile'
]) ).
command(name('rastrepl'),
  description(['magnify','a','rast
er','image','by','a','factor','o
f','two']) ).
command(name('rcp'),
  description(['remote','file','co
py']) ).
command(name('rdist'),
  description(['remote','file','di
stribution','program']) ).
command(name('red'),
  description(['basic','line','edi
tor']) ).
command(name('refer'),
  description(['expand','and','ins
ert','references','from','a','bi
bliographic','database']) ).
command(name('remove'),
  description(['filters','provided
','with','textedit']) ).
command(name('reset'),
  description(['establish','or','r
estore','terminal','characterist
ics']) ).
command(name('rev'),
  description(['reverse','the','or
der','of','characters','in','eac
h','line']) ).
command(name('ring'),
  description(['sunview','programm
able','alarms']) ).
command(name('rlogin'),
  description(['remote','login'])
  ).
command(name('rm'),
  description(['remove','files','o
r','directories']) ).
command(name('rmdel'),
  description(['remove','a','delta
','from','an','sccs','file']) ).
command(name('rmdir'),
  description(['remove','files','o
r','directories']) ).
command(name('roffbib'),
  description(['format','and','prin
nt','a','bibliographic','databas
e']) ).
command(name('rpcgen'),
  description(['rpc','protocol','c
ompiler']) ).
command(name('rsh'),
  description(['remote','shell'])
  ).
command(name('rup'),
  description(['show','host','stat
us','of','local','machines']) ).
command(name('ruptime'),
  description(['show','host','stat
us','of','local','machines']) ).
command(name('rusers'),
  description(['who','s','logged',
'in','on','local','machines'])
  ).
command(name('rwall'),
  description(['write','to','all',
'users','over','a','network'])
  ).
command(name('rwho'),
  description(['who','s','logged',
'in','on','local','machines'])
  ).
command(name('sact'),
  description(['show','editing','a
ctivity','status','of','an','scc
s','file']) ).
command(name('sccs'),
  description(['front','end','for'
,'the','source','code','control'
,'system']) ).
command(name('sccs-admin'),
  description(['create','and','adm
inister','sccs','history','files
']) ).
command(name('sccs-cdc'),
  description(['change','the','del
ta','commentary','of','an','sccs
','delta']) ).
command(name('sccs-comb'),
  description(['combine','sccs','d
eltas']) ).
command(name('sccs-comb'),
  description(['make','a','delta',
'to','an','sccs','file']) ).
command(name('sccsdiff'),
  description(['compare','two','ve
rsions','of','an','sccs','file']
) ).
command(name('sccs-get'),
  description(['retrieve','a','ver
sion','of','an','sccs','file'])
  ).
command(name('sccs-help'),
  description(['ask','for','help',
'regarding','sccs','error','or',
'warning','messages']) ).
command(name('sccs-prs'),
  description(['display','selected
','portions','of','an','sccs','h
istory']) ).
command(name('sccs-prt'),
  description(['display','delta','
table','information','from','an'
,'sccs','file']) ).
command(name('sccs-rmdel'),
  description(['remove','a','delta
','from','an','sccs','file']) ).
command(name('sccs-show'),
  description(['show','editing','a
ctivity','status','of','an','scc
s','file']) ).
command(name('sccs-sccsdiff'),
  description(['compare','two','ve
rsions','of','an','sccs','file']
) ).
command(name('sccs-unget'),
  description(['undo','a','previou
s','get','of','an','sccs','file'
]) ).
command(name('sccs-val'),
  description(['validate','an','sc
cs','file']) ).
command(name('screenblank'),
  description(['turn','off','the',
'screen','when','the','mouse','a
nd','keyboard','are','idle']) ).
command(name('screendump'),
  description(['dump','a','frame',
'buffer','image','to','a','file'
]) ).
command(name('screenload'),
  description(['load','a','frame',
'buffer','image','from','a','fil
e']) ).
command(name('script'),
  description(['make','typescript'
,'of','a','terminal','session']
) ).
command(name('scrolldefaults'),
  description(['edit','default','s
ettings','for','sunview','utilit
ies']) ).
command(name('sdiff'),
  description(['contrast','two','t
ext','files','by','displaying','
them','side','by','side']) ).
command(name('sed'),
  description(['stream','editor'])
  ).
command(name('selection'),
  description(['sunview','selectio
n','service']) ).
command(name('set'),
  description(['sunview','programm
able','alarms']) ).
command(name('sh'),
  description(['standard','unix','
system','shell','and','command',
'level','language']) ).
command(name('shelltool'),
  description(['run','a','shell','
in','a','sunview','terminal','wi
ndow']) ).
command(name('shift'),
  description(['filters','provided
','with','textedit']) ).
command(name('size'),
  description(['display','the','si
ze','of','an','object','file'])
  ).
command(name('sleep'),
  description(['suspend','executio
n','for','a','specified','interv
al']) ).
command(name('snap'),
  description(['sunview','applicat
ion','for','system','and','netwo
rk','administration']) ).
command(name('soelim'),
  description(['resolve','and','el
iminate','requests','from','nrof
f','or','troff','input']) ).
command(name('sort'),
  description(['sort','and','colla
te','lines']) ).
command(name('sortbib'),
  description(['sort','a','bibliog
raphic','database']) ).
command(name('sparc'),
  description(['return','a','true'
,'exit','status','if','the','pro
cessor','is','of','indicated','t
ype']) ).
command(name('spell'),
  description(['report','spelling'
,'errors']) ).
command(name('spellin'),
  description(['report','spelling'
,'errors']) ).
command(name('spline'),
  description(['interpolate','smoo
th','curve']) ).
command(name('split'),
  description(['split','a','file',
'into','pieces']) ).
command(name('strings'),
  description(['find','printable',
'strings','in','an','object','fi
le','or','binary']) ).
command(name('strip'),
  description(['remove','symbols',
'and','relocation','bits','from'
,'an','object','file']) ).
command(name('stty'),
  description(['set','or','alter',
'the','options','for','a','termi
nal']) ).
command(name('stty_from_defaults')
  ,
  description(['edit','default','s
ettings','for','sunview','utilit
ies']) ).
command(name('stty_from_defaults')
  ,
  description(['set','terminal','e
diting','characters','from','the
','defaults','database']) ).
command(name('su'),
  description(['super','user',',',
'temporarily','switch','to','a',
'new','user','id']) ).
command(name('sum'),
  description(['calculate','a','ch
ecksum','for','a','file']) ).
command(name('sun'),
  description(['return','a','true'
,'exit','status','if','the','pro
cessor','is','of','indicated','t
ype']) ).
command(name('sunview'),
  description(['the','sunview','wi
ndow','environment']) ).
command(name('sv_acquire'),
  description(['change','owner',',
','group',',','mode','of','windo
w','devices']) ).
command(name('sv_release'),
  description(['change','owner',',
','group',',','mode','of','windo
w','devices']) ).
command(name('swin'),
  description(['set','or','get','s
unview','user','input','options'
]) ).
command(name('switcher'),
  description(['switch','between',
'multiple','desktops','on','the'
,'same','physical','screen']) ).
command(name('symorder'),
  description(['rearrange','a','li
st','of','symbols']) ).
command(name('sync'),
  description(['update','the','sup
```

```
  er','block']) ).
command(name('sysex'),
  description(['start','the','syst
  em','exerciser']) ).
command(name('syswait'),
  description(['execute','a','comm
  and',',','suspending','terminati
  on','until','user','input']) ).
command(name('t300'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('t300s'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('t401s'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('t450'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('tabs'),
  description(['set','tab','stops'
  ,'on','a','terminal']) ).
command(name('tail'),
  description(['display','the','la
  st','part','of','a','file']) ).
command(name('talk'),
  description(['talk','to','anothe
  r','user']) ).
command(name('tar'),
  description(['create','tape','ar
  chives',',','and','add','or','ex
  tract','files']) ).
command(name('tbl'),
  description(['format','tables','
  for','nroff','or','troff']) ).
command(name('tcopy'),
  description(['copy','a','magneti
  c','tape']) ).
command(name('tcov'),
  description(['construct','test',
  'coverage','analysis']) ).
command(name('tee'),
  description(['replicate','the','
  standard','output']) ).
command(name('tek'),
  description(['graphics','filters
  ','for','various','plotters']
  ) ).
command(name('tektool'),
  description(['sunview','tektroni
  x','terminal','emulator','window
  ']) ).
command(name('telnet'),
  description(['user','interface',
  'to','a','remote','system','usin
  g','the','telnet','protocol'])
  ).
command(name('test'),
  description(['return','true','or
  ','false','according','to','a','
  conditional','expression']) ).
command(name('textedit'),
  description(['sunview','window',
  'and','mouse','based','text','ed
  itor']) ).
command(name('textedit_filters'),
  description(['filters','provided
  ','with','textedit']) ).
command(name('tftp'),
  description(['trivial','file','t
  ransfer','program']) ).
command(name('time'),
  description(['time','a','command
  ']) ).
command(name('tip'),
  description(['connect','to','rem
  ote','system']) ).
command(name('toolplaces'),
  description(['display','current'
  ,'window','locations',',','sizes
  ',',','and','other','attributes'
  ]) ).
command(name('touch'),
  description(['update','the','acc
  ess','and','modification','times
  ','of','a','file']) ).
command(name('tput'),
  description(['initialize','a','t
  erminal','or','query','the','ter
  minfo','database']) ).
command(name('tr'),
  description(['translate','charac
  ters']) ).
command(name('trace'),
  description(['trace','system','c
  alls','and','signals']) ).
command(name('traffic'),
  description(['sunview','program'
  ,'to','display','ethernet','traf
  fic']) ).
command(name('troff'),
  description(['typeset','or','for
  mat','commands']) ).
command(name('true'),
  description(['provide','truth','
  values']) ).
command(name('tset'),
  description(['establish','or','r
  estore','terminal','characterist
  ics']) ).
command(name('tsort'),
  description(['topological','sort
  ']) ).
command(name('tty'),
  description(['display','the','na
  me','of','the','terminal']) ).
command(name('u3b2'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('u3b5'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('u3b15'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('ul'),
  description(['do','underlining']
  ) ).
command(name('uname'),
  description(['display','the','na
  me','of','the','current','system
  ']) ).
command(name('uncompress'),
  description(['compress','or','ex
  pand','files',',','display','exp
  anded','contents']) ).
command(name('unexpand'),
  description(['expand','tab','cha
  racters','to','space','character
  s',',','and','vice','versa']) ).
command(name('unget'),
  description(['undo','a','previou
  s','get','of','an','sccs','file'
  ]) ).
command(name('unifdef'),
  description(['resolve','and','re
  move','ifdef','ed','lines','from
  ','cpp','input']) ).
command(name('uniq'),
  description(['remove','or','repo
  rt','adjacent','duplicate','line
  s']) ).
command(name('units'),
  description(['conversion','progr
  am']) ).
command(name('unix'),
  description(['convert','text','f
  ile','from','iso','format','to',
  'dos','format']) ).
command(name('unload'),
  description(['unload','optional'
  ,'clusters']) ).
command(name('unloadc'),
  description(['unload','optional'
  ,'clusters']) ).
command(name('unpack'),
  description(['compress','and','e
  xpand','files']) ).
command(name('unwhiteout'),
  description(['remove','a','tfs',
  'whiteout','entry']) ).
command(name('uptime'),
  description(['show','how','long'
  ,'the','system','has','been','up
  ']) ).
command(name('users'),
  description(['display','a','comp
  act','list','of','users','logged
  ','in']) ).
command(name('ustar'),
  description(['process','tape','a
  rchives']) ).
command(name('uucp'),
  description(['system','to','syst
  em','copy']) ).
command(name('uudecode'),
  description(['encode','a','binar
  y','file',',','or','decode','its
  ','ascii','representation']) ).
command(name('uuencode'),
  description(['encode','a','binar
  y','file',',','or','decode','its
  ','ascii','representation']) ).
command(name('uulog'),
  description(['system','to','syst
  em','copy']) ).
command(name('uuname'),
  description(['system','to','syst
  em','copy']) ).
command(name('uupick'),
  description(['public','system','
  to','system','file','copy']) ).
command(name('uusend'),
  description(['send','a','file','
  to','a','remote','host']) ).
command(name('uustat'),
  description(['uucp','status','in
  quiry','and','job','control'])
  ).
command(name('uuto'),
  description(['public','system','
  to','system','file','copy']) ).
command(name('uux'),
  description(['remote','system','
  command','execution']) ).
command(name('vacation'),
  description(['reply','to','mail'
  ,'automatically']) ).
command(name('val'),
  description(['validate','an','sc
  cs','file']) ).
command(name('vax'),
  description(['return','a','true'
  ,'exit','status','if','the','pro
  cessor','is','of','the','indicat
  ed','type']) ).
command(name('vedit'),
  description(['visual','display',
  'editor','based','on','ex']) ).
command(name('vfontinfo'),
  description(['inspect','and','pr
  int','out','information','about'
  ,'fonts']) ).
command(name('vgrind'),
  description(['grind','nice','pro
  gram','listings']) ).
command(name('vi'),
  description(['visual','display',
  'editor','based','on','ex']) ).
command(name('view'),
  description(['visual','display',
  'editor','based','on','ex']) ).
command(name('vplot'),
  description(['plot','graphics','
  for','a','versatec','printer']
  ).
command(name('vswap'),
  description(['convert','a','fore
  ign','font','file']) ).
command(name('vtroff'),
  description(['troff','to','a','r
  aster','plotter']) ).
command(name('vwidth'),
  description(['make','a','troff',
  'width','table','for','a','font'
  ]) ).
command(name('w'),
  description(['who','is','logged'
  ,'in',',','and','what','are','th
  ey','doing']) ).
command(name('wait'),
  description(['wait','for','a','p
  rocess','to','finish']) ).
command(name('wall'),
  description(['write','to','all',
  'users','logged','in']) ).
command(name('wc'),
  description(['display','a','coun
  t','of','lines',',','words','and
  ','characters']) ).
command(name('what'),
  description(['extract','sccs','v
  ersion','information','from','a'
  ,'file']) ).
command(name('whatis'),
  description(['display','a','one'
  ,'line','summary','about','a','k
  eyword']) ).
command(name('whereis'),
  description(['locate','the','bin
  ary',',','source',',','and','man
  ual','page','for','a','command']
  ) ).
command(name('which'),
  description(['locate','a','comma
  nd',',','display','its','pathnam
```

```
  e','or','alias']) ).
command(name('who'),
  description(['who','is','logged'
  ,'in','on','the','system']) ).
command(name('whoami'),
  description(['display','the','ef
  fective','current','username'])
  ).
command(name('whois'),
  description(['tcp','ip','interne
  t','user','name','directory','se
  rvice']) ).
command(name('write'),
  description(['write','a','messag
  e','to','another','user']) ).
command(name('xargs'),
  description(['construct','the','
  arguments','list','and','execute
  ','a','command']) ).

command(name('xget'),
  description(['send','or','receiv
  e','secret','mail']) ).
command(name('xsend'),
  description(['send','or','receiv
  e','secret','mail']) ).
command(name('xstr'),
  description(['extract','strings'
  ,'from','c','programs','to','imp
  lement','shared','strings']) ).
command(name('yacc'),
  description(['yet','another','co
  mpiler','compiler',',','parsing'
  ,'program','generator']) ).
command(name('yes'),
  description(['be','repetitively'
  ,'affirmative']) ).
command(name('ypcat'),
  description(['print','values','i
  n','a','nis','data','base']) ).
command(name('ypmatch'),
  description(['print','the','valu
  e','of','one','or','more','keys'
  ,'from','a','nis','map']) ).
command(name('yppasswd'),
  description(['change','your','ne
  twork','password','in','the','ni
  s','database']) ).
command(name('ypwhich'),
  description(['return','hostname'
  ,'of','nis','server','or','map',
  'master']) ).
command(name('zcat'),
  description(['compress','or','ex
  pand','files',',','display','exp
  anded','contents']) ).
```

A.4.2 Consultas

```
query(treference(1),
  words(['compare','two','files'])
  ).
query(treference(2),
  words(['join','columns','of','tw
  o','different','files','into','a
  ','single','file']) ).
query(treference(3),
  words(['delete','a','file']) ).
query(treference(4),
  words(['change','the','permissio
  n','access','to','a','file']) ).
query(treference(5),
  words(['rename','a','directory']
  ) ).
query(treference(6),
  words(['overwrite','a','file'])
  ).
query(treference(7),
  words(['put','a','file','in','an
  other','directory']) ).
query(treference(8),
  words(['get','the','name','of','
  the','current','directory']) ).
query(treference(9),
  words(['create','a','new','direc
  tory']) ).
query(treference(10),
  words(['become','the','root','us
  er']) ).
query(treference(11),
  words(['get','information','abou
  t','the','type','of','a','file']
  ) ).
query(treference(12),
  words(['distinguish','between','
  ascii','and','binary','files'])
  ).
query(treference(13),
  words(['change','my','password']
  ) ).
query(treference(14),
  words(['display','when','a','fil
  e','was','last','written','to'])
  ).
query(treference(15),
  words(['set','time','and','date'
  ,'of','the','clock']) ).
query(treference(16),
  words(['copy','a','file','from',
  'a','remote','host']) ).
query(treference(17),
  words(['format','c','source','co
  de']) ).
query(treference(18),
  words(['stop','a','process']) ).
query(treference(19),
  words(['split','files','accordin
  g','to','a','pattern','parameter
  ']) ).
query(treference(20),
  words(['send','binary','files','
  via','electronic','mail']) ).
query(treference(21),
  words(['locate','a','regular','e
  xpression','in','a','file']) ).
query(treference(22),
  words(['look','for','a','pattern
  ','in','a','file']) ).

/*
Datos para la evaluacion:
  Documentos relevantes a las
  consultas
*/

query_rel(treference(1),
  docs([diff,cmp]) ).
query_rel(treference(2),
  docs([join]) ).
query_rel(treference(3),
  docs([rm]) ).
query_rel(treference(4),
  docs([chmod]) ).
query_rel(treference(5),
  docs([mv]) ).
query_rel(treference(6),
  docs([mv]) ).
query_rel(treference(7),
  docs([mv]) ).
query_rel(treference(8),
  docs([pwd]) ).
query_rel(treference(9),
  docs([mkdir]) ).
query_rel(treference(10),
  docs([su]) ).
query_rel(treference(11),
  docs([file]) ).
query_rel(treference(12),
  docs([file]) ).
query_rel(treference(13),
  docs([passwd]) ).
query_rel(treference(14),
  docs([ls]) ).
query_rel(treference(15),
  docs([date]) ).
query_rel(treference(16),
  docs([rcp,ftp]) ).
query_rel(treference(17),
  docs([cb]) ).
query_rel(treference(18),
  docs([kill]) ).
query_rel(treference(19),
  docs([csplit]) ).
query_rel(treference(20),
  docs([uuencode]) ).
query_rel(treference(21),
  docs([grep,egrep,fgrep,awk]) ).
query_rel(treference(22),
  docs([grep,egrep,fgrep,awk]) ).
```

A.5 Valores de similitud obtenidos en los experimentos

A continuación se presentan las bases de datos en forma de hechos Prolog correspondientes a los valores de la similitud entre las descripciones de las órdenes y las consultas calculados por los tres sistemas. Se incluyen sólo los nombres de las órdenes que resultan con similitud no nula, y aparecen ordenadas decrecientemente por su valor de similitud. La estructura de los predicados es la siguiente:

```
query_doc_rel(
    reference(NumeroConsulta),
    docRel([ doc(NombreOrden,Similitud),
             doc(NombreOrden,Similitud), ... ]).
```

A.5.1 Sistema Ares

```
query_doc_rel_sem(reference(1),
  docRel([doc('sccs-
  sccsdiff',88.39483268996321),doc
  (sccsdiff,88.39483268996321),doc
  (dircmp,50.82670281569073),doc(a
  cctcom,4.598410478965118),doc(ad
  min,4.598410478965118),doc(ar,4.
  598410478965118),doc(bar,4.59841
  0478965118),doc(checknr,4.598410
  478965118),doc(compress,4.598410
  478965118),doc(cp,4.598410478965
  118),doc(crontab,4.59841047896511
  18),doc(crypt,4.598410478965118)
  ,doc(csplit,4.598410478965118),d
  oc(ctags,4.598410478965118),doc(
  dd,4.598410478965118),doc(dos2un
  ix,4.598410478965118),doc(du,4.5
  98410478965118),doc(egrep,4.5984
  10478965118),doc(fgrep,4.5984104
  78965118),doc(grep,4.59841047896
  5118),doc(indent,4.598410478965
  118),doc(install,4.59841047896511
  8),doc(make,4.598410478965118),d
  oc(mv,4.598410478965118),doc(pac
  k,4.598410478965118),doc(pcat,4.
  598410478965118),doc(pr,4.598410
  478965118),doc(rm,4.598410478965
  118),doc(rmdir,4.598410478965118
  ),doc('sccs-
  admin',4.598410478965118),doc('s
  ccs-
  val',4.598410478965118),doc(spli
  t,4.598410478965118),doc(tar,4.5
  98410478965118),doc(uncompress,4
  .598410478965118),doc(unix,4.598
  410478965118),doc(unpack,4.59841
  0478965118),doc(uudecode,4.59841
  0478965118),doc(uuencode,4.59841
  0478965118),doc(uusend,4.5984104
  78965118),doc(val,4.598410478965
  118),doc(vswap,4.598410478965118
  ),doc(zcat,4.598410478965118)])
  .
query_doc_rel_sem(reference(2),
  docRel([doc(paste,80.51652498380
  913),doc(adjacentscreens,62.1263
  5494153508),doc(ld,62.1263549415
  3508),doc('ld.so',62.12635494153
  508),doc('sccs-
  sccsdiff',37.568129874272 47),doc
  (sccsdiff,37.568129874272 47),doc
  (chgrp,4.598410478965118),doc(ch
  mod,4.598410478965118),doc(cpio,
  4.598410478965118),doc(delta,4.5
  98410478965118),doc(df,4.5984104
  78965118),doc(error,4.5984104789
  65118),doc(get,4.598410478965118
  ),doc(head,4.598410478965118),do
  c(paxcpio,4.598410478965118),doc
  (sact,4.598410478965118),doc('sc
  sc-
  comb',4.598410478965118),doc('sc
  sc-
  get',4.598410478965118),doc('scc
  s-
  show',4.598410478965118),doc(scr
  eendump,4.598410478965118),doc(s
  ize,4.598410478965118),doc(strin
  gs,4.598410478965118),doc(tail,4
  .598410478965118),doc(touch,4.59
  8410478965118)])).
query_doc_rel_sem(reference(3),
  docRel([doc(crontab,46.490903385
  73001),doc(rm,46.490903385730 01)
  ,doc(rmdir,46.490903385730 01),do
  c(colrm,41.89249290676489),doc(i
  perm,41.89249290676489),doc(lprm
  ,41.89249290676489),doc(rmdel,41
  .89249290676489),doc('sccs-
  rmdel',41.89249290676489),doc(st
  rip,41.89249290676489),doc(unifd
  ef,41.89249290676489),doc(uniq,4
  1.89249290676489),doc(unwhiteout
  ,41.89249290676489),doc(acctcom,
  4.598410478965118),doc(admin,4.5
  98410478965118),doc(ar,4.5984104
  78965118),doc(bar,4.598410478965
  118),doc(checknr,4.598410478965
  118),doc(compress,4.598410478965
  118),doc(cp,4.598410478965118),do
  c(crypt,4.598410478965118),doc(c
  split,4.598410478965118),doc(cta
  gs,4.598410478965118),doc(dd,4.5
  98410478965118),doc(dos2unix,4.5
  98410478965118),doc(du,4.5984104
  78965118),doc(egrep,4.5984104789
  65118),doc(fgrep,4.5984104789651
  18),doc(grep,4.598410478965118),
  doc(indent,4.598410478965118),do
  c(install,4.598410478965118),doc
  (make,4.598410478965118),doc(mv,
  4.598410478965118),doc(pcat,4.59
  8410478965118),doc(pack,4.59841
  0478965118),doc(pr,4.598410478965
  118),doc('sccs-
  admin',4.598410478965118),doc('s
  ccs-
  val',4.598410478965118),doc(spli
  t,4.598410478965118),doc(tar,4.5
  98410478965118),doc(uncompress,4
  .598410478965118),doc(unix,4.598
  410478965118),doc(unpack,4.59841
  0478965118),doc(uudecode,4.59841
  0478965118),doc(uuencode,4.59841
  0478965118),doc(uusend,4.5984104
  78965118),doc(val,4.598410478965
  118),doc(vswap,4.598410478965118
  ),doc(zcat,4.598410478965118)])
  .
query_doc_rel_sem(reference(4),
  docRel([doc(chmod,102.2476724796
  7037),doc(lockscreen,59.50961491
  0496644),doc(touch,59.5096149104
  96644),doc(mv,44.697854826201 63)
  ,doc(coloredit,36.568129355328639
  ),doc(env,36.568129355328639),doc
  (atty,36.568129355328639),doc(dd,
  30.298129399902123),doc(dos2unix
  ,30.298129399902123),doc('old-
  sun3cvt',30.298129399902123),doc
  (ranlib,30.298129399902123),doc(
  unix,30.298129399902123),doc(vsw
  ap,30.298129399902123),doc(cd,26
  .309557974826365),doc(cdc,26.309
  557974826365),doc(chfn,26.309557
  974826365),doc(chgrp,26.30955797
  4826365),doc(chkey,26.3095579748
  26365),doc(chsh,26.3095579748263
  65),doc(passwd,26.30955797482636
  5),doc('sccs-
  cdc',26.309557974826365),doc(sv_
  acquire,26.309557974826365),doc(
  sv_release,26.309557974826365),d
  oc(yppasswd,26.309557974826365),
  doc(delta,4.598410478965118),doc
  ('sccs-
  comb',4.598410478965118),doc(scr
  eendump,4.598410478965118),doc(s
  trings,4.598410478965118),doc(su
  m,4.598410478965118)])).
query_doc_rel_sem(reference(5),
  docRel([doc(mv,75.93811450484401
  ),doc(cd,73.774957162962 65),doc(
  coloredit,62.126354941535 08),doc
  (env,62.126354941535 08),doc(atty
  ,62.126354941535 08),doc(dd,51.47
  412171341257),doc(dos2unix,51.47
  412171341257),doc('old-
  sun3cvt',51.47412171341257),doc(
  ranlib,51.47412171341257),doc(un
  ix,51.47412171341257),doc(vswap,
  51.47412171341257),doc(cdc,44.69
  7854826201 63),doc(chfn,44.697854
  826201 63),doc(chgrp,44.69785482
  620163),doc(chkey,44.697854826201
  63),doc(chmod,44.697854826201 63)
  ,doc(chsh,44.697854826201 63),doc
  (passwd,44.697854826201 63),doc('
  sccs-
  cdc',44.697854826201 63),doc(sv_a
  cquire,44.697854826201 63),doc(sv
  _release,44.697854826201 63),doc(
  yppasswd,44.697854826201 63),doc(
  dircmp,29.077102336761 02),doc(mk
  dir,29.077102336761 02),doc(rm,29
  .077102336761 02),doc(rmdir,29.07
  7102336761 02)])).
query_doc_rel_sem(reference(6),
  docRel([doc(write,62.126354941535
  08),doc(uusend,52.913005857191 2
  66),doc(enroll,48.314595378226 15
  ),doc(kill,48.31459537822615),do
  c(lpr,48.31459537822615),doc(mai
  l,48.31459537822615),doc(xget,48
  .31459537822615),doc(xsend,48.31
  459537822615),doc(crontab,46.490
  90338573001),doc(rm,46.490903385
  73001),doc(rmdir,46.490903385730
  01),doc(colrm,41.89249290676489)
  ,doc(ipcrm,41.89249290676489),do
  c(lprm,41.89249290676489),doc(rm
  del,41.89249290676489),doc('sccs
  -
  rmdel',41.89249290676489),doc(st
  rip,41.89249290676489),doc(unifd
  ef,41.89249290676489),doc(uniq,4
  1.89249290676489),doc(unwhiteout
  ,41.89249290676489),doc(acctcom,
  4.598410478965118),doc(ar,4.5984104
  78965118),doc(bar,4.598410478965
  118),doc(checknr,4.598410478965
  118),doc(compress,4.598410478965
  118),doc(cp,4.598410478965118),do
  c(crypt,4.598410478965118),doc(c
  split,4.598410478965118),doc(cta
  gs,4.598410478965118),doc(dd,4.5
  98410478965118),doc(dos2unix,4.5
  98410478965118),doc(du,4.5984104
  78965118),doc(egrep,4.5984104789
  65118),doc(fgrep,4.5984104789651
  18),doc(grep,4.598410478965118),
  doc(indent,4.598410478965118),do
  c(install,4.598410478965118),doc
  (make,4.598410478965118),doc(mv,
  4.598410478965118),doc(pack,4.59
  8410478965118),doc(pcat,4.598410
  478965118),doc('sccs-
  admin',4.598410478965118),doc('s
  ccs-
  val',4.598410478965118),doc(spli
  t,4.598410478965118),doc(tar,4.5
  98410478965118),doc(uncompress,4
  .598410478965118),doc(unix,4.598
  410478965118),doc(unpack,4.59841
  0478965118),doc(uudecode,4.59841
  0478965118),doc(uuencode,4.59841
  0478965118),doc(val,4.598410478965
  118),doc(vswap,4.598410478965118
  ),doc(zcat,4.598410478965118)]
  ]).
query_doc_rel_sem(reference(7),
  docRel([doc(write,50.826702815690
  73),doc(date,48.314595378226 15)
  ,doc(domainname,48.314595378226 1
  5),doc(hostname,48.314595378226 1
  5),doc(stty,48.314595378226 15),d
  oc(stty_from_defaults,48.3145953
  7822615),doc(swin,48.314595378226
  15),doc(tabs,48.314595378226 15)
  ,doc(cd,29.077102336761 02),doc(d
  ircmp,29.077102336761 02),doc(mkd
  ir,29.077102336761 02),doc(rm,29.
  077102336761 02),doc(rmdir,29.077
  102336761 02),doc(banner,14.87078
  3815255679),doc(comm,14.870783815
  255679),doc(getopt,14.870783815255
  679),doc(getoptcvt,14.8707838152
```

```
5679),doc(getopts,14.87078381525
679),doc(rev,14.87078381525679),
doc(users,14.870783815256791,doc
(acctcom,4.598410478965118),doc(
admin,4.598410478965118),doc(ar,
4.598410478965118),doc(bar,4.598
410478965118),doc(checknr,4.5984
10478965118),doc(chgrp,4.5984104
78965118),doc(chmod,4.5984104789
65118),doc(compress,4.5984104789
65118),doc(cp,4.598410478965118)
,doc(cpio,4.598410478965118),doc
(crontab,4.598410478965118),doc(
crypt,4.598410478965118),doc(csp
lit,4.598410478965118),doc(ctags
,4.598410478965118),doc(dd,4.598
410478965118),doc(df,4.598410478
965118),doc(dos2unix,4.598410478
965118),doc(du,4.598410478965118
),doc(egrep,4.598410478965118),d
oc(error,4.598410478965118),doc(
fgrep,4.598410478965118),doc(get
,4.598410478965118),doc(grep,4.5
98410478965118),doc(head,4.59841
0478965118),doc(indent,4.5984104
78965118),doc(install,4.59841047
8965118),doc(make,4.598410478965
118),doc(mv,4.598410478965118),d
oc(pack,4.598410478965118),doc(p
aste,4.598410478965118),doc(paxc
pio,4.598410478965118),doc(pcat,
4.598410478965118),doc(pr,4.5984
10478965118),doc(sact,4.59841047
8965118),doc('sccs-
admin',4.598410478965118),doc('s
ccs-
get',4.598410478965118),doc('scc
s-
sccsdiff',4.598410478965118),doc
('sccs-
show',4.598410478965118),doc('sc
cs-
val',4.598410478965118),doc(sccs
diff,4.598410478965118),doc(size
,4.598410478965118),doc(split,4.
598410478965118),doc(tail,4.5984
10478965118),doc(tar,4.598410478
965118),doc(touch,4.598410478965
118),doc(uncompress,4.598410478
65118),doc(unix,4.598410478965118
),doc(unpack,4.598410478965118)
,doc(uudecode,4.598410478965118)
,doc(uuencode,4.598410478965118)
,doc(uusend,4.598410478965118),d
oc(val,4.598410478965118),doc(vs
wap,4.598410478965118),doc(zcat,
4.598410478965118)]}).
query_doc_rel_sem(reference(8),
 docRel([doc(pwd,71.60124932341586
),doc(logname,66.30305090551194
),doc(id,61.479012002933231,doc(
domainname,55.877767693495021,do
c(hostname,55.877767693495021,do
c(uname,55.877767693495021,doc(e
nv,55.7042524700732826),doc(graph
,55.7042524700732826),doc(enroll,
45.57266378384907),doc(xget,45.5
7266378384907),doc(xsend,45.5726
6378384907),doc(gcore,40.8617433
219492234),doc(swin,40.8617433219
49234),doc(look,39.180343030761925
),doc(lookbib,39.180343030761925
),doc(lorder,39.180343030761925
),doc(strings,39.180343030761925
),doc(which,37.2259485687509161
),doc(delta,36.527290210812381,do
c(mkdir,36.527290210812381),doc('
old-
syslog',36.527290210812381),doc('
sccs-
comb',36.527290210812381),doc(scr
ipt,36.527290210812381),doc(vwidt
h,36.527290210812381),doc(tty,30.
7395060014666161,doc(du,29.07710
233676102),doc(ls,29.077102336761
02),doc(arch,25.1382616920283971
),doc(defaults_from_input,25.138
2616920283971),doc(hostid,25.1382
616920283971),doc(input,25.138261
6920283971),doc(mach,25.138261692
028397),doc(mps,25.1382616920283
97),doc(ps,25.1382616920283971),d
oc(toolplaces,25.138261692028397
),doc(whoami,25.1382616920283971)
)}).
query_doc_rel_sem(reference(9),
 docRel([doc(mkdir,59.10441474925
973),doc(delta,30.027312412498713
),doc('old-
syslog',30.027312412498713),doc('
sccs-
comb',30.027312412498713),doc(scr
ipt,30.027312412498713),doc(vwidt
h,30.027312412498713),doc(cd,29.
077102336761021),doc(dircmp,29.
077102336761021),doc(rm,29.077102
33676102),doc(rmdir,29.077102336
76102),doc(addbib,27.613083499451
13),doc(admin,27.613083499451131
),doc(ar,27.613083499451131),doc(
bar,27.613083499451131),doc(chkey
,27.613083499451131),doc(ctags,27
.613083499451131),doc(iconedit,27
.613083499451131),doc(indxbib,27.
613083499451131),doc('sccs-
admin',27.613083499451131),doc(ta
r,27.613083499451131)}]}).
query_doc_rel_sem(reference(10),
 docRel([doc(gcore,55.704252470073826),doc(logname,55.704252470073826),doc(swin,55.704252470073826),doc(error,51.474121771341257),doc(indent,51.474121771341257),doc(finger,22.224122379287476)]}).
query_doc_rel_sem(reference(11),
 docRel([doc(env,55.704252470073826),doc(graph,55.704252470073826),doc(enroll,45.572663783849071,doc(xget,45.572663783849071),doc(xsend,45.572663783849071),doc(gcore,40.861743321949234),doc(logname,40.861743321949234),doc(swin,40.861743321949234),doc(look,39.180343030761925),doc(lookbib,39.180343030761925),doc(lorder,39.180343030761925),doc(strings,39.180343030761925),doc(delta,36.527290210812381),doc(mkdir,36.527290210812381),doc('old-
syslog',36.527290210812381),doc('sccs-
comb',36.527290210812381),doc(script,36.527290210812381),doc(vwidth,36.527290210812381),doc(chfn,30.7395060014666161,doc(chsh,30.7395060014666161),doc(finger,30.7395060014666161),doc(lpstat,30.7395060014666161),doc(passwd,30.7395060014666161),doc(prt,30.7395060014666161),doc('sccs-
prt',30.7395060014666161),doc(vfontinfo,30.7395060014666161),doc(what,30.7395060014666161),doc(chgrp,4.598410478965118),doc(chmod,4.598410478965118),doc(cpio,4.598410478965118),doc(df,4.598410478965118),doc(error,4.598410478965118),doc(get,4.598410478965118),doc(head,4.598410478965118),doc(paste,4.598410478965118),doc(paxcpio,4.598410478965118),doc(sact,4.598410478965118),doc('sccs-
get',4.598410478965118),doc('sccs-
sccsdiff',4.598410478965118),doc('sccs-
show',4.598410478965118),doc(sccsdiff,4.598410478965118),doc(size,4.598410478965118),doc(tail,4.598410478965118),doc(touch,4.598410478965118)]}).
query_doc_rel_sem(reference(13),
 docRel([doc(yppasswd,71.3906743329478),doc(look,51.7953653738169),doc(wc,51.795365373816),doc(mv,44.697854826201163),doc(coloredit,36.568129352386391),doc(env,36.568129352386391),doc(stty,36.568129352386391),doc(dd,30.298129399902123),doc(dos2unix,30.298129399902123),doc('old-
sunlcvt',30.298129399902123),doc(ranlib,30.298129399902123),doc(unix,30.298129399902123),doc(vswap,30.298129399902123),doc(cd,26.309557974826365),doc(cdc,26.309557974826365),doc(chfn,26.309557974826365),doc(chgrp,26.309557974826365),doc(chkey,26.309557974826365),doc(chmod,26.309557974826365),doc(chsh,26.309557974826365),doc(passwd,26.309557974826365),doc('sccs-
cdc',26.309557974826365),doc(sv_acquire,26.309557974826365),doc(sv_release,26.309557974826365)}]).
query_doc_rel_sem(reference(15),
 docRel([doc(date,90.249120911963
26),doc(pr,48.314595378226151),do
c(foption,42.770271973488531),doc
(clock,37.568129874272471),doc(to
uch,37.568129874272471),doc(domai
nname,30.739506001466616),doc(ho
stname,30.739506001466616),doc(s
tty,30.739506001466616),doc(stty
_from_defaults,30.739506001466616
),doc(swin,30.739506001466616),
doc(tabs,30.739506001466616)]}).
query_doc_rel_sem(reference(16),
 docRel([doc(cp,29.736672170993515),doc(dd,29.736672170993515),doc(on,29.077102336761021),doc(cpio,25.138261692028397),doc(get,25.138261692028397),doc(paxcpio,25.138261692028397),doc(tcopy,25.138261692028397),doc(acctcom,4.598410478965118),doc(admin,4.598410478965118),doc(ar,4.598410478965118),doc(bar,4.598410478965118),doc(checknr,4.598410478965118),doc(compress,4.598410478965118),doc(crontab,4.598410478965118),doc(crypt,4.598410478965118),doc(csplit,4.598410478965118),doc(ctags,4.598410478965118),doc(dos2unix,4.598410478965118),doc(du,4.598410478965118),doc(egrep,4.598410478965118),doc(fgrep,4.598410478965118),doc(grep,4.598410478965118),doc(indent,4.598410478965118),doc(install,4.598410478965118),doc(make,4.598410478965118),doc(mv,4.598410478965118),doc(pack,4.598410478965118),doc(pcat,4.598410478965118),doc(pr,4.598410478965118),doc(rm,4.598410478965118),doc(rmdir,4.598410478965118),doc('sccs-
admin',4.598410478965118),doc('sccs-
val',4.598410478965118),doc(split,4.598410478965118),doc(tar,4.598410478965118),doc(uncompress,4.598410478965118),doc(unix,4.598410478965118),doc(unpack,4.598410478965118),doc(uudecode,4.598410478965118),doc(uuencode,4.598410478965118),doc(uusend,4.598410478965118),doc(val,4.598410478965118),doc(vswap,4.598410478965118)]}).
query_doc_rel_sem(reference(17),
 docRel([doc(indent,65.630880687886598),doc(error,34.891374686399374),doc(fdformat,30.739506001466616),doc(nroff,30.739506001466616),doc(roffbib,30.739506001466616),doc(tbl,30.739506001466616),doc(troff,30.739506001466616),doc(ctrace,27.613083499451131),doc(cxref,27.613083499451131)]}).
query_doc_rel_sem(reference(18),
 docRel([doc(kill,105.01521684160502),doc(checknr,75.938114504844401)]}).
query_doc_rel_sem(reference(20),
 docRel([doc(uudecode,49.679526837086541,doc(uuencode,49.679526837086541),doc(write,39.527056898463841,doc(uusend,35.337916480431731,doc(enroll,30.739506001466616),doc(kill,30.739506001466616),doc(lpr,30.739506001466616),doc(mail,30.739506001466616),doc(xget,30.739506001466616),doc(xsend,30.739506001466616),doc(acctcom,4.598410478965118),doc(admin,4.598410478965118),doc(ar,4.598410478965118),doc(bar,4.598410478965118),doc(checknr,4.598410478965118),doc(compress,4.598410478965118),doc(cp,4.598410478965118),doc(crontab,4.598410478965118),doc(crypt,4.598410478965118),doc(csplit,4.598410478965118),doc(ctags,4.598410478965118),doc(dd,4.598410478965118),doc(dos2unix,4.598410478965118),doc(du,4.598410478965118),doc(egrep,4.598410478965118),doc(fgrep,4.598410478965118),doc(grep,4.598410478965118),doc(indent,4.598410478965118),doc(install,4.598410478965118),doc(make,4.598410478965118),doc(mv,4.598410478965118),doc(pack,4.598
```

```
4104789651181),doc(pcat,4.598410478965118),doc(pr,4.5984104789651181),doc(rm,4.5984104789651181),doc(rmdir,4.5984104789651181),doc('sccs-admin',4.5984104789651181),doc('sccs-val',4.5984104789651181),doc(split,4.5984104789651181),doc(tar,4.5984104789651181),doc(uncompress,4.5984104789651181),doc(unix,4.5984104789651181),doc(unpack,4.5984104789651181),doc(val,4.5984104789651181),doc(vswap,4.5984104789651181),doc(zcat,4.5984104789651181)]}).
query_doc_rel_sem(reference(21),
  docRel([doc(apropos,50.826702815 69073),doc(whereis,50.8267028156 9073),doc(which,50.82670281569073),doc(acctcom,47.86778147913074},doc(egrep,47.86778147913074),doc(fgrep,47.86778147913074),doc(grep,47.86778147913074),doc(delta,4.5984104789651181),doc(prt,4.5984104789651181),doc(rmdel,4.598410478965118),doc('sccs-comb',4.5984104789651181),doc('sccs-prt',4.5984104789651181),doc('sccs-rmdel',4.5984104789651181),doc(screendump,4.5984104789651181],doc(screenload,4.5984104789651181),doc(strings,4.5984104789651181),doc(strip,4.5984104789651181),doc(what,4.5984104789651181)]}).
query_doc_rel_sem(reference(22),
  docRel([doc(acctcom,58.509614389 51056),doc(egrep,58.509614389510 56),doc(fgrep,58.509614389510 56),doc(grep,58.509614389510 56),doc(lptest,50.826702815690 73),doc(delta,4.5984104789651181),doc(prt,4.5984104789651181),doc(rmdel,4.598410478965118),doc('sccs-comb',4.5984104789651181),doc('sccs-prt',4.5984104789651181),doc('sccs-rmdel',4.5984104789651181),doc(screendump,4.5984104789651181),doc(screenload,4.5984104789651181),doc(strings,4.5984104789651181),doc(strip,4.5984104789651181),doc(what,4.5984104789651181)]}).
```

A.5.2 Sistema MV

```
query_doc_rel_bp(reference(1),
  docRel([doc('sccs-sccsdiff',0.8036215864306 51),doc(sccsdiff,0.8036215864306 51),doc(dircmp,0.5896319882187374),doc(comm,0.26040388694306854),doc(cmp,0.24089113888406724),doc(sdiff,0.2333518961335278),doc(rastrepl,0.2288405128842072),doc(cp,0.08744517607255535),doc(rm,0.06327915745120452),doc(rmdir,0.06327915745120452),doc(mkstr,0.06295802849562036),doc(rcp,0.06217882530460605),doc(install,0.05955604356657378),doc(pack,0.05866181448390061),doc(pcat,0.05866181448390061),doc(unpack,0.05866181448390061),doc(rmdel,0.05522443949153294),doc('sccs-rmdel',0.05522443949153294),doc('sccs-val',0.05324076760495028),doc(val,0.05324076760495028),doc(ftp,0.052765629642732265),doc(delta,0.05199732387613294),doc('sccs-comb',0.05199732387613294),doc(usend,0.04785126618738668),doc(size,0.04734459865218338),doc(crypt,0.04626087661071179),doc(organizer,0.04554594991829247),doc(get,0.04444525746431452),doc('sccs-get',0.04444525746431452),doc(uupick,0.04357134430903718),doc(uuto,0.04357134430903718),doc(tail,0.04292905260274094),doc(rdist,0.04228441938890436 4),doc(what,0.04228006201614117),doc(cpio,0.04213372817323721),doc(paxcpio,0.04213372817323721),doc(prt,0.04211409762360289),doc('sccs-prt',0.04211409762360289),doc(split,0.04029456462425825 4),doc(chgrp,0.04005343061976332 5),doc(compress,0.03989923826349539),doc(uncompress,0.03989923826349539),doc(zcat,0.03989923826349539),doc(admin,0.03856204156121814),doc('sccs-admin',0.03856204156121814),doc(mv,0.03812062911060251),doc(sum,0.03812062911060251),doc(tftp,0.037983321605069 15),doc(chmod,0.03797551468934572),doc(dd,0.037662223194007576),doc(vswap,0.037638501354597315),doc(egrep,0.03591748837239999),doc(fgrep,0.03591748837239999),doc(grep,0.03591748837239999),doc(more,0.03553087 99028616 55),doc('sccs-unget',0.03548421117611330 6),doc(unget,0.03548421117611330 6),doc('old-ccat',0.03533946208972064),doc('old-compact',0.03533946208972064),doc('old-uncompact',0.03533946208972064),doc(find,0.03522646762639984),doc(acctcom,0.03517568391589695),doc(sact,0.03503703123928749 36),doc('sccs-show',0.03503703123928749 36),doc(indent,0.03447279631974537),doc(bar,0.03444343229845 95),doc(ar,0.03358153207495854),doc(file,0.03348746841871725),doc(screendump,0.03327699247351697),doc(screenload,0.03327699247351697),doc(cluster,0.03292201722961564 4),doc(diff,0.03288925968400564),doc(touch,0.03250713283346281),doc(cut,0.03240013301631917),doc(head,0.03218262586813667),doc('old-filemerge',0.03212629384003774),doc(checknr,0.03191369201434858 6),doc(dos2unix,0.03112279352556239 58),doc(unix,0.03122793525562395 8),doc(ln,0.03078350570383538),doc(pg,0.03068913304369809 4),doc(make,0.03040268954208803),doc(objdump,0.03033335923181670 2),doc(diffmk,0.03031690115590284 3),doc(strings,0.03012758032827709 4),doc(strip,0.03006655018460062 9),doc(df,0.02971716992020351 2),doc(uudecode,0.02947852126047083),doc(uuencode,0.02947852126047083),doc(ctags,0.02901854027800787 3),doc(csplit,0.02872936782681763 7),doc(crontab,0.02858212559205839 8),doc(paste,0.02701383298039478),doc(error,0.02640315348240331),doc(du,0.02629878814535703 2),doc(pr,0.02521571152392239 3)]}).
query_doc_rel_bp(reference(2),
  docRel([doc(paste,0.263864492989 82856),doc('sccs-sccsdiff',0.21321381927821 97),doc(sccsdiff,0.21321381927821 97),doc(pr,0.19877217010798368),doc(colrm,0.18831995930175335),doc(cmp,0.14095094920739922),doc(comm,0.13738588960617615),doc(sdiff,0.13653956476662507),doc(rastrepl,0.12073344146127872),doc(cp,0.09226999986465836),doc(rm,0.06677061116114365),doc(rmdir,0.06677061116114365),doc(mkstr,0.06643176378248776),doc(rcp,0.06560956773281915),doc(install,0.06284207292655941),doc(pack,0.06189850438403 8926),doc(pcat,0.06189850438403 8926),doc(unpack,0.06189850438403 8926),doc(rmdel,0.05827147080339 4246),doc('sccs-rmdel',0.05827147080339 4246),doc('sccs-val',0.05617834899922929),doc(val,0.05617834899922929),doc(ftp,0.05567699510323 88),doc(delta,0.05486629774788 88),doc('sccs-comb',0.05486629774788 88),doc(usend,0.05049147960969823),doc(size,0.04995685648347974),doc(crypt,0.04881333962970907),doc(organizer,0.04805896656105 59),doc(get,0.04689754294524307),doc('sccs-get',0.04689754294524307),doc(uupick,0.04597541126982329),doc(uuto,0.04597541126982329),doc(tail,0.04529768085272340 6),doc(rdist,0.04461747973443528),doc(what,0.04461288194182592),doc(cpio,0.04458474068557526),doc(paxcpio,0.04458474068557526),doc(prt,0.04437760395220086),doc('sccs-prt',0.04437760395220086),doc(split,0.04251783390935002),doc(chgrp,0.04226339523632761),doc(compress,0.04210069525296696),doc(uncompress,0.04210069525296696),doc(zcat,0.04210069525296696),doc(admin,0.04068971816904217 6),doc('sccs-admin',0.04068971816904217 6),doc(mv,0.04022395060371909),doc(sum,0.04022395060371909),doc(tftp,0.04007906710123972),doc(chmod,0.04007082943419209),doc(dd,0.03974025195025896),doc(vswap,0.03971522124917543),doc(egrep,0.03789925066318606 4),doc(fgrep,0.03789925066318606 4),doc(grep,0.03789925066318606 4),doc(more,0.03749131091 12285054),doc(page,0.03749131091 12285054),doc('sccs-unget',0.03744206721821751),doc(unget,0.03744206721821751),doc('old-ccat',0.03728933154105701 4),doc('old-compact',0.03728933154105701 4),doc('old-uncompact',0.03728933154105701 4),doc(find,0.03717010256144278 5),doc(acctcom,0.03711651683868845),doc(sact,0.03697021399978901),doc('sccs-show',0.03697021399978901),doc(indent,0.03637484712842776),doc(bar,0.03634386293510674),doc(tar,0.03634386293510674),doc(ar,0.03543440700994745),doc(file,0.03533650231321198),doc(screendump,0.03511306430991696),doc(screenload,0.03511306430991696),doc(cluster,0.03473850315997363 5),doc(diff,0.03470393820322299),doc(touch,0.03430072734550094),doc(cut,0.03418782376922501 6),doc(head,0.03395831557401706 6),doc('old-filemerge',0.03389887540294632 4),doc(checknr,0.03369905293428649 6),doc(dos2unix,0.03295094951171 7086),doc(unix,0.03295094951171 7086),doc(ln,0.03248199840342772),doc(pg,0.03238241866468416),doc(make,0.03208017052427696),doc(objdump,0.03200701488544183),doc(diffmk,0.03198964872837578),doc(strings,0.03178988204570696 4),doc(strip,0.03172437835747637),doc(df,0.03135682707345803),doc(uudecode,0.03110501087512405 2),doc(uuencode,0.03110501087512405 2),doc(ctags,0.03061965025151593),doc(csplit,0.03031452259061955 3),doc(crontab,0.03015936723968199 2),doc(error,0.02785995840669951 ),doc(du,0.02774979246195797]]).
query_doc_rel_bp(reference(3),
  docRel([doc(keylogout,0.58225809 90530402),doc(cp,0.0939647819551
```

```
9971),doc(rm,0.06799702967351379),doc(rmdir,0.06799702967351379),doc(mkstr,0.06765195846837459),doc(rcp,0.0668146606180997 4),doc(install,0.063996333465029 61),doc(pack,0.06303543379571377),doc(pcat,0.06303543379571377),doc(unpack,0.06303543379571377),doc(rmdel,0.05934178016994841),doc('sccs-
rmdel',0.05934178016994841),doc('sccs-
val',0.0572102126591375),doc(val,0.0572102126591375),doc(ftp,0.05669965007554 33),doc(delta,0.055874062123458 58),doc('sccs-
comb',0.055874062123458 58),doc(usend,0.05141888890300014),doc(size,0.05087444601195 1985),doc(crypt,0.04970992545289416),doc(organizer,0.04894169633316 97204),doc(get,0.04775894010564813),doc('sccs-
get',0.04775894010564813),doc(uupick,0.04681987104807907),doc(uuto,0.04681987104807907),doc(tail,0.04612969232302 6425),doc(rdist,0.04543699751583 69),doc(what,0.04543231527263 0586),doc(cpio,0.04527507129121612),doc(paxcpio,0.04527507129121612),doc(prt,0.04525397715659491),doc('sccs-
prt',0.04525397715659491),doc(split,0.04329878615324157),doc(chgrp,0.04303967404240826),doc(compress,0.04287398564441 3),doc(uncompress,0.04287398564441 3),doc(zcat,0.04287398564441 3),doc('sccs-sccsdiff',0.04270091583984958 5),doc(admin,0.04143709224212355),doc('sccs-
admin',0.04143709224212355),doc(mv,0.04096276962608818),doc(sum,0.04096276962608818),doc(tftp,0.04081522495567149),doc(chmod,0.04080683598212 5526),doc(dd,0.04047018657080247),doc(vswap,0.04044469611417725),doc(egrep,0.03859537043519343),doc(fgrep,0.03859537043519343),doc(grep,0.03859537043519343),doc(more,0.03817993779300786),doc(page,0.03817993779300786),doc('sccs-
unget',0.03812978960954034 6),doc(unget,0.03812978960954034 6),doc('old-
ccat',0.03797424853852899 4),doc('old-
compact',0.03797424853852899 4),doc('old-
uncompact',0.03797424853852899 4),doc(find,0.03785282960400397),doc(acctcom,0.03779825963801384 4),doc(sact,0.03764926956131744),doc('sccs-
show',0.03764926956131744),doc(indent,0.03704296720591286),doc(bar,0.03701141390610334 5),doc(tar,0.03701141390610334 5),doc(ar,0.03608525342240543 5),doc(file,0.03598555044749884),doc(screendump,0.03575800841551 75),doc(screenload,0.03575800841551 75),doc(cluster,0.03537656746141 6084),doc(diff,0.03534136762800186),doc(touch,0.03493075073861 86),doc(cut,0.03481577339015982),doc(head,0.03458204967116904),doc('old-
filemerge',0.03452151772445666),doc(checknr,0.03431802498874 63),doc(dos2unix,0.03355618067223192),doc(unix,0.03355618067223192),doc(ln,0.03307861603906085),doc(pg,0.03297720725557843 5),doc(make,0.03266940753030065),doc(objdump,0.03259490819506688),doc(diffmk,0.03257722306268902),doc(strings,0.03237378713761671),doc(strip,0.03230708030125773 5),doc(df,0.03193277796777 09),doc(uudecode,0.03167633650029559 6),doc(uuencode,0.03167633650029559 6),doc(ctags,0.03118206094775044 5),doc(csplit,0.03087132881983 51),doc(crontab,0.03071332362349954 8),doc(paste,0.02902788969379 75),doc(error,0.02837168007810038),doc(du,0.02825949064500492 6),doc(cmp,0.02822863238452704),doc(sdiff,0.02734515278440 46),doc(pr,0.02709570660 85)]]).
query_doc_rel_bp(reference(4),
docRel([doc(chmod,0.659722649714 8246),doc(touch,0.33883649982041 864),doc(lockscreen,0.2513404531 4930825),doc(chgrp,0.20128402152 815544),doc(cd,0.19030129185001 62),doc(chkey,0.16749025236391 96),doc(cdc,0.15957209641487802),doc('sccs-
cdc',0.15957209641487802),doc(chfn,0.15764167688392242),doc(chsh,0.15764167688392242),doc(passwd,0.15764167688392242),doc(yppasswd,0.14466617196701254),doc(sv_acquire,0.12965974598173 79),doc(sv_release,0.12965974598173 79),doc(cp,0.06537966837050642),doc(rm,0.04731159012696255),doc(rmdir,0.04731159012696255),doc(mkstr,0.04707149335161 0085),doc(rcp,0.04688910659974854),doc(install,0.04452794942755 0088),doc(pack,0.04385936593903545),doc(pcat,0.04385936593903545),doc(unpack,0.04385936593903545),doc(rmdel,0.04128936211309996),doc('sccs-
rmdel',0.04128936211309996),doc('sccs-
val',0.039806240734362466),doc(val,0.039806240734362466),doc(ftp,0.03945099686848841),doc(delta,0.03887656179403979),doc('sccs-
comb',0.03887656179403979),doc(usend,0.03577670095654422),doc(size,0.03539788354301178),doc(crypt,0.03458762286472769),doc(organizer,0.03405309743795268 5),doc(get,0.03323014858186841 6),doc('sccs-
get',0.03323014858186841 6),doc(uupick,0.03257675459442594 5),doc(uuto,0.03257675459442594 5),doc(tail,0.03209653578029796),doc(rdist,0.03161456630171667),doc(what,0.03161130845006 4594),doc(cpio,0.03150189980627 04),doc(paxcpio,0.03150189980627 04),doc(prt,0.03148722273793265 5),doc('sccs-
prt',0.03148722273793265 5),doc(split,0.03012682220551631 7),doc(compress,0.02983125065399527 8),doc(uncompress,0.02983125065399527 8),doc(zcat,0.02983125065399527 8),doc('sccs-
sccsdiff',0.02971083057540988 4),doc(sccsdiff,0.02971083057540988 4),doc(admin,0.02883147592807463 7),doc('sccs-
admin',0.02883147592807463 7),doc(mv,0.02850144743557189),doc(sum,0.02850144743557189),doc(tftp,0.02839878746636898),doc(dd,0.02815871329001816),doc(vswap,0.02814097754785982),doc(egrep,0.02685423695116828),doc(fgrep,0.02685423695116828),doc(grep,0.02685423695116828),doc(more,0.02656518345887 8743),doc(page,0.02656518345887 8743),doc('sccs-
unget',0.02653029090087 7038),doc(unget,0.02653029090087 7038),doc('old-
ccat',0.02642206712353071 4),doc('old-
compact',0.02642206712353071 4),doc('old-
uncompact',0.02642206712353071 4),doc(find,0.02633758515584 0608),doc(acctcom,0.02629961596988 4053),doc(sact,0.02619595030278815),doc('sccs-
show',0.02619595030278815),doc(indent,0.02577409175000023),doc(bar,0.02575213730883026),doc(tar,0.02575213730883026),doc(ar,0.02510772496601 3146),doc(file,0.02503835272569 8113),doc(screendump,0.02488003146658 6544),doc(screenload,0.02488003146658 6544),doc(cluster,0.02461462901938 0628),doc(diff,0.02459013735997981),doc(cut,0.02422443463135671),doc(head,0.02406181222398 1842),doc('old-
filemerge',0.02401969475700241 3),doc(checknr,0.02387810673540148 2),doc(dos2unix,0.02334802378594 6935),doc(unix,0.02334802378594 6935),doc(ln,0.02301573953335224 5),doc(pg,0.02294518041007243 7),doc(make,0.02273103672507831 7),doc(objdump,0.02267918090180434 7),doc(diffmk,0.02266687577991528),doc(strings,0.02252532728659859 7),doc(strip,0.02247891339887948 5),doc(df,0.02221847792581525),doc(uudecode,0.02204004875538683),doc(uuencode,0.02204004875538683),doc(ctags,0.02169613722772045 4),doc(csplit,0.02147990231055 5446),doc(crontab,0.02136999492022 01307),doc(paste,0.02019728841541 8575),doc(error,0.01974070493245 7526),doc(du,0.01966264475099555 68),doc(cmp,0.01964117422370114 2),doc(sdiff,0.01902645846012386),doc(pr,0.01885289653705454)]]).
query_doc_rel_bp(reference(5),
docRel([doc(mv,0.592395448566659 3),doc(rm,0.37653353114365513),doc(rmdir,0.37653353114365513),doc(mkdir,0.36114187667769015),doc(ircmp,0.31742510901016 12),doc(ls,0.29001360193961534),doc(organizer,0.27101462855681 75),doc(cd,0.26471072724692 49),doc(pwd,0.21939867921303 58),doc(du,0.15648692085485952),doc(whois,0.15060863287219659)]]).
query_doc_rel_bp(reference(6),
docRel([doc(cp,0.093964781955199 71),doc(rm,0.06799702967351379),doc(rmdir,0.06799702967351379),doc(mkstr,0.06765195846837459),doc(rcp,0.06681466061809974),doc(install,0.06399633346502961),doc(pack,0.06303543379571377),doc(pcat,0.06303543379571377),doc(unpack,0.06303543379571377),doc(rmdel,0.05934178016994841),doc('sccs-
rmdel',0.05934178016994841),doc('sccs-
val',0.05721021265913 75),doc(val,0.05721021265913 75),doc(ftp,0.05669965007554 33),doc(delta,0.05587406212345858),doc('sccs-
comb',0.05587406212345858),doc(usend,0.05141888890300014),doc(size,0.05087444601195 1985),doc(crypt,0.04970992545289416),doc(organizer,0.04894169633316 97204),doc(get,0.04775894010564813),doc('sccs-
get',0.04775894010564813),doc(uupick,0.04681987104807907),doc(uuto,0.04681987104807907),doc(tail,0.04612969232302 6425),doc(rdist,0.04543699751583 69),doc(what,0.04543231527263 0586),doc(cpio,0.04527507129121612),doc(paxcpio,0.04527507129121612),doc(prt,0.04525397715659491),doc('sccs-
prt',0.04525397715659491),doc(split,0.04329878615324157),doc(chgrp,0.04303967404240826),doc(compress,0.04287398564441 3),doc(uncompress,0.04287398564441 3),doc(zcat,0.04287398564441 3),doc('sccs-
sccsdiff',0.04270091583984958 5),doc(admin,0.04143709224212355),doc('sccs-
admin',0.04143709224212355),doc(mv,0.04096276962608818),doc(sum,0.04096276962608818),doc(tftp,0.04081522495567149),doc(chmod,0.04080683598212 5526),doc(dd,0.04047018657080247),doc(vswap,0.04044469611417725),doc(egrep,0.03859537043519343),doc(fgrep,0.03859537043519343),doc(grep,0.03859537043519343),doc(more,0.03817993779300786),doc(page,0.03817993779300786),doc('sccs-
unget',0.03812978960954034 6),doc(unget,0.03812978960954034 6),doc('old-
ccat',0.03797424853852899 4),doc('old-
compact',0.03797424853852899 4),doc('old-
uncompact',0.03797424853852899 4),doc(find,0.03785282960400397),doc(acctcom,0.03779825963801384 4)
```

```
,doc(sact,0.037649269561317441),doc('sccs-
show',0.037649269561317441),doc(indent,0.037042967205912186),doc(bar,0.037011413906103345),doc(tar,0.037011413906103345),doc(ar,0.036085253422405435),doc(file,0.035985550447498841),doc(screendump,0.03575800841551751),doc(screenload,0.03575800841551751),doc(cluster,0.035376567461416084),doc(diff,0.035341367628001861),doc(touch,0.034930750738618691),doc(cut,0.034815773390159821),doc(head,0.034582049671169041),doc('old-filemerge',0.034521517724466611),doc(checknr,0.034318024988746311),doc(dos2unix,0.033555180672231921),doc(unix,0.033555180672231921),doc(ln,0.033078616039060851),doc(pg,0.032977207255578435),doc(make,0.032669407530300651),doc(objdump,0.032594908195066881),doc(diffmk,0.032577223062689021),doc(strings,0.032373787137616711),doc(strip,0.032307080301257375),doc(df,0.03193277796777091),doc(uudecode,0.031676336500295596),doc(uuencode,0.031676336500295596),doc(ctags,0.031182060947750445),doc(csplit,0.030871328811983511),doc(crontab,0.030713323623499548),doc(paste,0.029027889693979751),doc(error,0.028371680078100381),doc(du,0.028259490645004926),doc(cmp,0.028228632834527041),doc(sdiff,0.027345152784404661),doc(pr,0.027095706608511)]}).
query_doc_rel_bp(reference(7),
  docRel([doc(rm,0.33751116357809666),doc(rmdir,0.33751116357809666),doc(talk,0.31449243134775307),doc(write,0.31446409709167417),doc(mkdir,0.2795112179387981),doc(dircmp,0.24567596436059302),doc(organizer,0.24292780075408535),doc(ls,0.22446041384811161),doc(yacc,0.20846654534294404),doc(cd,0.20487710344832769),doc(pwd,0.16980692634884081),doc(du,0.14026926787119864),doc(cpio,0.12992451507416616),doc(paxcpio,0.12992451507416616),doc(login,0.12950152364047657),doc(look,0.12861787679768105),doc(whois,0.11656582945592675),doc(gfxtool,0.098189328268934311),doc(shelltool,0.093257506111343941),doc(clock,0.093098506333389831),doc(strings,0.092902086623222411),doc(newgrp,0.090400691617055761),doc(who,0.088684792913005851),doc(lookbib,0.087657383468926241),doc(wall,0.083178078668091041),doc(users,0.082473408297050641),doc(getopt,0.081270412316791331),doc(getoptcvt,0.081270412316791331),doc(getopts,0.081270412316791331),doc(ypcat,0.080245493192603381),doc(yppasswd,0.079652867277600571),doc(pr,0.077755737126453521),doc(banner,0.077486865585591511),doc(rev,0.075843173945116721),doc(comm,0.075072818024414571),doc(inline,0.073323344751275611),doc(rusers,0.073220634880046911),doc(rwho,0.073220634880046911),doc(logname,0.070084536968957921),doc(lastcomm,0.069685806720616721),doc(cp,0.063687885328647641),doc(perfmeter,0.057374255440967161),doc(w,0.055650491050278251),doc(mkstr,0.045853457897092831),doc(rcp,0.045285950280253603),doc(install,0.043375731442851),doc(pack,0.042724448412353531),doc(pcat,0.042724448412353531),doc(unpack,0.042724448412353531),doc(rmdel,0.040220946742189011),doc('sccs-rmdel',0.040220946742189011),doc('sccs-val',0.038776203037430331),doc(val,0.038776203037430331),doc(ftp,0.038430151563671271),doc(delta,0.037870580735888561),doc('sccs-comb',0.037870580735888561),doc(uusend,0.034850932786095086),doc(size,0.034481917760554946),doc(crypt,0.033692623619868405),doc(get,0.032370275730714375),doc('sccs-
get',0.032370275730714375),doc(upick,0.031733778915340656),doc(uuto,0.031733789153406561),doc(tail,0.031265996618982521),doc(rdist,0.030796498722748696),doc(what,0.030793325170251691),doc(prt,0.030672450335579881),doc('sccs-prt',0.030672450335579881),doc(split,0.029347251917341231),doc(chgrp,0.029171629710184045),doc(compress,0.029059328660022074),doc(uncompress,0.029059328660022074),doc(zcat,0.029059328660022074),doc('sccs-sccsdiff',0.028942024605912296),doc(sccsdiff,0.028942024605912296),doc(admin,0.028085424391525691),doc('sccs-admin',0.028085424391525691),doc(mv,0.027763935810907824),doc(sum,0.027763935810907824),doc(tftp,0.027663932300498661),doc(chmod,0.027658246381135761),doc(dd,0.027430007058515943),doc(vswap,0.027412793545350031),doc(egrep,0.026159349017222455),doc(fgrep,0.026159349017222455),doc(grep,0.026159349017222455),doc(more,0.025877775155965524),doc(page,0.025877775155965524),doc('sccs-unget',0.025843785487797771),doc(unget,0.025843785487797771),doc('old-sccat',0.025738362139939682),doc('old-compact',0.025738362139939682),doc('old-uncompact',0.025738362139939682),doc(find,0.025656066251864895),doc(acctcom,0.025619079567449195),doc(sact,0.025518096382874035),doc('sccs-show',0.025518096382874035),doc(indent,0.025107153962936855),doc(bar,0.025085767621180383),doc(tar,0.025085767621180383),doc(ar,0.024458030276886861),doc(file,0.024390453132549915),doc(screendump,0.024232628613972681),doc(screenload,0.024232628613972681),doc(cluster,0.023977693822330608),doc(diff,0.023953835916129738),doc(touch,0.023675552609822436),doc(cut,0.023597596218100731),doc(head,0.023439181872811591),doc('old-filemerge',0.023398154249481995),doc(checknr,0.023260230000118592),doc(dos2unix,0.022743863629028888),doc(unix,0.022743863629028888),doc(ln,0.022420177659013805),doc(pg,0.022351444343835941),doc(make,0.022142822419751372),doc(objdump,0.022092327915100782),doc(diffmk,0.022080341204128152),doc(strip,0.021897242591494472),doc(df,0.021643546221457692),doc(uudecode,0.021469734135330956),doc(uuencode,0.021469734135330956),doc(ctags,0.021134721760947351),doc(csplit,0.020924112294093203),doc(crontab,0.020817018805276345),doc(paste,0.019674657590203951),doc(error,0.019229888792341776),doc(cmp,0.019132933568738728),doc(sdiff,0.018534124366487051)]}).
query_doc_rel_bp(reference(8),
  docRel([doc(uname,0.55838850572278881),doc(logname,0.39831152337665121),doc(hostname,0.38682354434736781),doc(pwd,0.37173380577030521),doc(tty,0.34969985220913941),doc(domainname,0.34907543594957181),doc(id,0.34711625803859725),doc(rm,0.34400018501874651),doc(rmdir,0.34400185018746551),doc(mkdir,0.32994000130602561),doc(dircmp,0.29000026760908654),doc(whois,0.28305944255483817),doc('sccs-unget',0.27108220654248577),doc(unget,0.27108220654248577),doc(swin,0.26839418505850341),doc(ls,0.26495705533512561),doc(organizer,0.24759955207241681),doc(ps,0.24755459134426722),doc(cd,0.24184056825267417),doc(nm,0.23184511906237441),doc(gcore,0.22476931527246957),doc(mach,0.20565446507277457),doc(find,0.20244867809410472),doc(arch,0.18649016485078126),doc(whoami,0.16456431387728661),doc(mps,0.15915015149520575),doc(defaults_from_input,0.15761154453999096),doc(input,0.15761154453999096),doc(hostid,0.14436870913502906),doc(du,0.14296679007765101),doc(toolplaces,0.12509931762863661)]}).
query_doc_rel_bp(reference(9),
  docRel([doc(newgrp,0.4443519249773744),doc(rm,0.35795388654117911),doc(rmdir,0.35795388654117911),doc(mkdir,0.34332171681208331),doc(dircmp,0.30176210631145001),doc(su,0.29893013237575267),doc(ls,0.27570319075908971),doc(organizer,0.25764170140910053),doc(cd,0.25164914456767201),doc(chkey,0.21033281782483321),doc(pwd,0.20857268595721287),doc(admin,0.20715250709988841),doc('sccs-admin',0.20715250709988841),doc(addbib,0.19635275135958458),doc(bar,0.18502765350753917),doc(tar,0.18502765350753917),doc(ar,0.18039756921425887),doc(mkstr,0.16910299502983561),doc(indxbib,0.16901529283314071),doc(ctags,0.15588550625426301),doc(du,0.14876524102042991),doc(whois,0.14317701087473471),doc(iconedit,0.13104454711216581)]}).
query_doc_rel_bp(reference(10),
  docRel([doc(su,0.18240638562675393),doc(finger,0.15898285536933257),doc(talk,0.13799005296119746),doc(write,0.13797762072084055),doc(last,0.12827500735237224),doc(groups,0.12586681609314884),doc(wall,0.12473965684640566),doc(swin,0.12409377677628632),doc(users,0.12368288839604497),doc(rwall,0.10807218624168337),doc(id,0.10666989494233982),doc(syswait,0.10594141242280068),doc(quota,0.10557635224639374),doc(crontab,0.10095790250392196),doc(telnet,0.096655505372590071),doc(whois,0.08940249810610521)]}).
query_doc_rel_bp(reference(11),
  docRel([doc(finger,0.624135556222626522),doc(vfontinfo,0.39341640812896231),doc('sccs-unget',0.27423770624707361),doc(unget,0.27423770624707361),doc(lpstat,0.26103478116465606),doc(whatis,0.25709271070107411),doc(what,0.25400231072012724),doc(prt,0.25300526064044915),doc('sccs-prt',0.25300526064044915),doc(swin,0.24405361713805754),doc(logname,0.20670601547080921),doc(gcore,0.20438507042145274),doc(chfn,0.19258230014224981),doc(chsh,0.19258230014224981),doc(passwd,0.19258230014224981),doc(mach,0.18700373881297985),doc(file,0.16929821274533172),doc(i,0.13505344286126533),doc(iapx,0.13505344286126533),doc(m,0.13505344286126533),doc(machid,0.13505344286126533),doc(pdp,0.13505344286126533),doc(sparc,0.13505344286126533),doc(sun,0.13505344286126533),doc(u3b15,0.13505344286126533),doc(u3b2,0.13505344286126533),doc(u3b5,0.13505344286126533),doc(vax,0.13505344286126533),doc(cp,0.06836041720676837),doc(rm,0.04946859047167924),doc(rmdir,0.04946859047167924),doc(mkstr,0.04921754735681729),doc(rcp,0.04860840391839152),doc(install,0.04655803977131251),doc(pack,0.04588974643703377),doc(pcat,0.04588974643703377),doc(unpack,0.04588974643703377),doc(rmdel,0.04317180081516197),doc('sccs-rmdel',0.04317180081516197),doc('sccs-val',0.041621061896692485),doc(val,0.041621061896692485),doc(ftp,0.041249621975283361),doc(delta,0.040648997617188211),doc('sccs-comb',0.040648997617188211),doc(uusend,0.037407809868525451),doc(size,0.037011721649057694),doc(cr
```

```
ypt,0.03616452007409947),doc(org
anizer,0.03560562489930046),doc(g
et,0.03474515667510155),doc('scc
s-
get',0.03474515667510155),doc(uu
pick,0.03406197355877191),doc(uu
to,0.03406197355877191),doc(tail
,0.03355986090648444),doc(rdist
,0.03305591778296706),doc(cpio,0
.03293811466523013),doc(paxcpio,
0.03293811466523013),doc(split,0
.03150034538908196),doc(chgrp,0
.03131183846502328),doc(compress
,0.03119129832458457),doc(uncom
press,0.03119129832458457),doc(
zcat,0.03119129832458457),doc('
sccs-
sccsdiff',0.03106538812929891),d
oc(sccsdiff,0.03106538812929891)
,doc(admin,0.03014594249638612 5)
,doc('sccs-
admin',0.03014594249638612 5),doc
(mv,0.02980086755183676 55),doc(su
m,0.02980086755183675 5),doc(tftp
,0.02969352717370307 7),doc(chmod
,0.02968742410274813 3),doc(dd,0.
02944250793559503 48),doc(vswap,0.
02942396334707957 8),doc(egrep,0.
02807855848010719),doc(fgrep,0.0
2807855848010719),doc(grep,0.028
07855848010719),doc(more,0.02777
632664228251),doc(page,0.0277763
2664228251),doc('old-
ccat',0.02762668543673139),doc('
old-
compact',0.02762668543673139),do
c('old-
uncompact',0.02762668543673139),
doc(find,0.02753835182015481),do
c(acctcom,0.02749865156688521),d
oc(sact,0.02739025963971094),doc
('sccs-
show',0.02739025963971094),doc(i
ndent,0.02694916797636082 5),doc(
bar,0.02692621260207803 5),doc(ta
r,0.02692621260207803 5),doc(ar,0
.02625242061588263 5),doc(screend
ump,0.02601434625723242 7),doc(sc
reenload,0.02601434625723242 7),d
oc(cluster,0.02573684374810563),
doc(diff,0.02571123548032981 3),d
oc(touch,0.02541250715588407 7),d
oc(cut,0.02532885986435183 5),doc
(head,0.02515882327608045),doc('
old-
filemerge',0.02511478561596125 5)
,doc(checknr,0.02496674240207908
),doc(dos2unix,0.02441249224324447
),doc(unix,0.02441249224324447
),doc(ln,0.02406505869540378 7),d
oc(pg,0.02399128268482788 8),doc(
make,0.02376735498342429 4),doc(o
bjdump,0.02371315589380542),doc(
diffmk,0.02370028976475395),doc
(strings,0.02355228788106739 4),d
oc(strip,0.02350375792048873),do
c(df,0.02323144883674285),doc(uu
decode,0.02304488483541216),doc(
uuencode,0.02304488483541216),do
c(ctags,0.02268529390906698 7),do
c(csplit,0.02245923284662866 6),d
oc(crontab,0.02234428232600972),d
oc(paste,0.02111811052174581),do
c(error,0.02064071077098455),doc(
du,0.02055909171868623 7),doc(cmp
,0.02053664232057705),doc(sdiff,
0.01989390082143728),doc(pr,0.01
9712425972024)}}}.
query_doc_rel_bp(reference(12),
docRel([doc(uudecode,0.467748889
9103856),doc(uuencode,0.46774888
99103856),doc(strings,0.23629568
657180725),doc(whereis,0.2155848
7157476156),doc(od,0.21406750107
107714),doc(cp,0.0634830351989647
1),doc(rm,0.04593910333604358 4)
,doc(rmdir,0.04593910333604358 4)
,doc(mkstr,0.04570597165621438),
doc(rcp,0.04514028941021759 6),do
c(install,0.04323621473311302),d
oc(pack,0.04258702653450755 6),do
c(pcat,0.04258702653450755 6),doc
(unpack,0.04258702653450755 6),do
c(rmdel,0.04009157730067608),doc
('sccs-
rmdel',0.04009157730067608),doc(
'sccs-
val',0.03865148056972963),doc(va
l,0.03865148056972963),doc(ftp,0
.03830654215992159),doc(delta,0.
03774877117454301),doc('sccs-
comb',0.03774877117454301),doc(u
usend,0.03473883582975933 26),doc(
size,0.03437100772973016),doc(cr
ypt,0.03358425229437396),doc(or
ganizer,0.03306523231781545),do
c(get,0.03226615773170897),doc('s
ccs-
get',0.03226615773170897),doc(uup
ick,0.03163171839395426),doc(uut
o,0.03163171839395426),doc(tail,
0.03116543049986876 2),doc(rdist,
0.03069744272921742 8),doc(what,0
.03069427938435469),doc(cpio,0.0
3058804463342726 7),doc(paxcpio,0
.03058804463342726 7),doc(prt,0.0
3057379333988104 5),doc('sccs-
prt',0.03057379333988104 5),doc(s
plit,0.02925285738170 73),doc(chg
rp,0.02907780005799264),doc(comp
ress,0.02896586022071394),doc(un
compress,0.02896586022071394),do
c(zcat,0.02896586022071394),doc(
'sccs-
sccsdiff',0.02884893471496183),
doc(sccsdiff,0.02884893471496183
),doc(admin,0.02799508848542494
8),doc('sccs-
admin',0.02799508848542494 8),doc
(mv,0.02767463962965725),doc(su
m,0.02767463962965725),doc(tftp
,0.02757495211078042),doc(chmod,
0.02756928448199631 7),doc(dd,0.0
2734184260447541 6),doc(vswap,0.0
2732462113573463 6),doc(egrep,0.0
2607520827348530 6),doc(fgrep,0.0
2607520827348530 6),doc(grep,0.02
6075208273485306),doc(more,0.025
79454008591649),doc(page,0.02579
454008591649),doc('sccs-
unget',0.02576065974447377 6),doc
(unget,0.02576065974447377 6),doc
('old-
ccat',0.02565557548758052 8),doc(
'old-
compact',0.02565557548758052 8),d
oc('old-
uncompact',0.02565557548758052 8)
,doc(find,0.02557354430170 55),do
c(acctcom,0.02553667658382574 2),
doc(sact,0.02543601820856634 6),d
oc('sccs-
show',0.02543601820856634 6),doc(
indent,0.02502639757231 82),doc(b
ar,0.02500508001907401 2),doc(tar
,0.02500508001907401 2),doc(ar,0.
02437936177269404),doc(file,0.02
43120019881690 8),doc(screendump,
0.02415827355369666),doc(screenl
oad,0.02415827355369666),doc(clu
ster,0.02390057030561066 5),doc(d
iff,0.02387678913776333 03),doc(to
uch,0.02359937449461912 3),doc(cu
t,0.02352169527366619),doc(head,
0.02336179046325968 3),doc('old-
filemerge',0.02332228948040221 03)
,doc(checknr,0.02318541418377627
3),doc(dos2unix,0.02267070868928
0766),doc(unix,0.02267070868928
0766),doc(ln,0.02234806384526001 3
),doc(pg,0.02227955160868218 3),d
oc(make,0.02207160071061744 7),do
c(objdump,0.02202126861999226),d
oc(diffmk,0.02200932046390781 4),
doc(strip,0.02182681078234555),d
oc(df,0.02157393041890145),doc(c
tags,0.02106674257666134),doc(cs
plit,0.02085681052869738),doc(cr
ontab,0.02075006150280456),doc(
paste,0.01961137465753909),doc(e
rror,0.01916803644872995),doc(d
u,0.01909224075225332 6),doc(cmp,
0.01907139307477279 5),doc(sdiff,
0.01847450992395309),doc(pr,0.01
8305982949956088)}}}.
query_doc_rel_bp(reference(13),
docRel([doc(chfn,0.6529757858140
959),doc(chsh,0.6529757858140959
),doc(passwd,0.6529757858140959)
,doc(yppasswd,0.5992292722846017
),doc(cd,0.2904957845703242),doc
(chgrp,0.2615474500496476 3),doc(
chkey,0.2556746294012836 6),doc(c
hmod,0.2479787343464263 4),doc(cd
c,0.2435875284551698),doc('sccs-
cdc',0.2435875284551698),doc(sv_
acquire,0.19792618993800 43),doc(
sv_release,0.19792618993800 43)]}
).
query_doc_rel_bp(reference(14),
docRel([doc(tail,0.4863253504547
362),doc(last,0.3273671003098439
5),doc(lastcomm,0.26670611671766
925),doc(size,0.10956514797912229
),doc(prt,0.09746069181060604),d
oc('sccs-
prt',0.09746069181060604),doc(co
mpress,0.09233505128451548),doc(
uncompress,0.09233505128451548),
doc(zcat,0.09233505128451548),do
c(tty,0.09119079509264143),doc(u
name,0.08010319883998218),doc(ps
,0.07893837465587829),doc(cal,0.
07779340047852808),doc(cat,0.077
79340047852808),doc(diff,0.07611
251772720849),doc(lpstat,0.07485
86145642824),doc(head,0.07447722
160459229),doc('old-
prmail',0.07360351479794632),doc
(date,0.07193276620208239),doc(c
p,0.07131606885340483),doc(prof,
0.06850564131343838),doc(nroff,0
.06849942226373383),doc(finger,0
.06745419110368109),doc(mach,0.0
6557757270998342),doc(lpq,0.0642
4029625404942),doc(du,0.06086071
725689547),doc(arch,0.0594666025
8930554),doc(clock,0.05923754066
0331996),doc(sdiff,0.05889156435
4944394),doc(man,0.05823152378304667),doc(pagesize,0.05715158748
020496),doc(basename,0.056831285
27627347),doc(dirname,0.05683128
527627347),doc(prs,0.05653138205
844571),doc('sccs-
prs',0.05653138205844571),doc(fo
ld,0.0563299881276 48),doc(pwd,0.
05525765399125959),doc(groups,0.
05340352106920892),doc(users,0.0
5247690934909272),doc(whoami,0.0
5247507114473963 5),doc(rm,0.0516
07429391315435),doc(rmdir,0.0516
07429391315435),doc(vedit,0.0515
47064413163285),doc(vi,0.0515470
64413163285),doc(view,0.05154706
4413163285),doc(traffic,0.051420
2241076020025),doc(mkstr,0.051345
53210051166),doc(wc,0.05106609177366576),doc(mps,0.05074862161533192 5),doc(rcp,0.05071005155239015 3),doc(which,0.05050143649305469),doc(domainname,0.05007635144
6073246),doc(banner,0.04930402788 68403114),doc(gprof,0.0492587524
6525941),doc(install,0.04857103724173491 4),doc(printenv,0.04821145723399377),doc(pack,0.04784174712311565),doc(pcat,0.04784174712311565),doc(unpack,0.04784174712311565),doc(comm,0.04776799634444483),doc(col,0.04506415844159017 4),doc(rmdel,0.04503838983526176),doc(
quota,0.04479456243199119 6),doc(
whatis,0.04459006618077697 4),doc
('sccs-
val',0.04342060270051217),doc(va
l,0.04342060270051217),doc(ftp,0
.04303310309047456),doc(delta,0
.04240650995620695),doc('sccs-
comb',0.04240650995620695),doc(a
tq,0.04219488606055915),doc(too
lplaces,0.01989074389830536),doc
(uusend,0.03902518523504126),doc
(from,0.03851287893192 88),doc(c
rypt,0.03772814018752744),doc(o
rganizer,0.03714508045214066),do
c(perfmeter,0.03650659847720476)
,doc(get,0.03624740876389128 5),d
oc('sccs-
get',0.03624740876389128 5),doc(u
upick,0.03553468733598835),doc(u
uto,0.03553468733598835),doc(rdi
st,0.03448513428781 35),doc(what
,0.03448157976567543),doc(cpio,0
.03436223693986348),doc(paxcpio,
0.03436223693986348),doc(split,0
.03286230383683185),doc(chgrp,0.
03265566457064726 5),doc('sccs-
sccsdiff',0.03240854063376713),d
oc(sccsdiff,0.03240854063376713)
,doc(admin,0.03144934157175089 5)
,doc('sccs-
admin',0.03144934157175089 5),doc
(mv,0.03108934686266899 6),doc(su
m,0.03108934686266899 6),doc(tftp
```

```
,0.030977365483524l3],doc{chmod,
0.0309709985383499481,doc{dd,0.0
3071549310218748811,doc{vswap,0.0
3069614671387090411,doc{egrep,0.0
2929257151569015711,doc{fgrep,0.0
2929257151569015711,doc{grep,0.02
9292571515690157]],doc{more,0.028
977272290836071,doc{page,0.02897
727229083607]],doc{'sccs-
unget',0.028939211527743494],doc
{unget,0.028939211527743494],doc
{'old-
ccat',0.028821161152922571,doc{'
old-
compact',0.028821161152922571,do
c{'old-
uncompact',0.028821161152922571,
doc{find,0.028729008317418491,do
c{acctcom,0.028687591572007051,d
oc{sact,0.028574513178729661,doc
{'sccs-
show',0.028574513178729661,doc{i
ndent,0.028114350342991071,doc{b
ar,0.028090402463212111,doc{tar,
0.028090402463212111,doc{ar,0.02
738747820318251,doc{file,0.02731
180704133466511,doc{screendump,0.
027139110389651261,doc{screenloa
d,0.027139110389651261,doc{clust
er,0.026849609698220131,doc{touc
h,0.026511249991131327],doc{cut,
0.026423986100036708],doc{'old-
filemerge',0.026200656072815231
,doc{checknr,0.026046212017787437
},doc{dos2unix,0.025467998131674
028],doc{ln,0.0251055428420165],
doc{pg,0.025028577029563271,doc{
make,0.024794967522424361,doc{ob
jdump,0.024738425064595881,doc{d
iffmk,0.024725002651516241,doc{s
trings,0.024570601713868781,doc
{strip,0.024519973497256146],doc
{df,0.024235890775714385],doc{uu
decode,0.024041260436870451,doc{
uuencode,0.024041260436870451,do
c{ctags,0.023666122128619271,doc
{csplit,0.023430287021803553],do
c{crontab,0.023310366465754264],
doc{paste,0.022031179526995657],
doc{error,0.021533138776411696],
doc{cmp,0.021424570791048521,doc
{pr,0.020564718375494638]]}}.
query_doc_rel_bp{reference{15},
docRel{[doc{from,0.63640234778320
59],doc{time,0.34703253762561561
},doc{date,0.223732524996981741,
doc{clock,0.218161759562621,doc{
at,0.18230998135743361,doc{batch
,0.18230998135743361,doc{stty,0.
18112627388922903],doc{touch,0.1
79416209326380531,doc{hostname,0
.16721883938004659],doc{swin,0.1
58660041516448l],doc{atq,0.15539
65692177971],doc{stty_from_defau
lts,0.15211817867802163],doc{dom
ainname,0.15090081797904495],doc
{tabs,0.15045501330480643],doc{p
rintenv,0.14528111818222111,doc{
iperm,0.11197663936883011]]}}.
query_doc_rel_bp{reference{16},
docRel{[doc{rcp,0.79224188219142
27],doc{uusend,0.7107945717692847
},doc{cp,0.56333204190589091,do
c{rsh,0.38231579340622746],doc{u
ucp,0.3480639387406121,doc{uulog
,0.3480639387406121,doc{uuname,0
.3480639387406121,doc{rlogin,0.3
3603589125330291,doc{mach,0.3307
5639361877346],doc{rdist,0.30848
270629720037],doc{cu,0.307910728
7973861],doc{tip,0.3079107287973
861],doc{arch,0.29993423361650511
],doc{uupick,0.28069169118979015
],doc{uuto,0.28069169118979015],
doc{uux,0.27996405765503491,doc{
hostname,0.27988472686144733],do
c{rup,0.2790786939990132],doc{ru
ptime,0.2790786939990132],doc{sp
io,0.27143040006282054],doc{paxc
pio,0.27143040006282054],doc{dd,
0.24262444250774895],doc{hostid,
0.23218982173813912],doc{tcopy,0
.23180084112069831,doc{on,0.2099
3249812168263],doc{pg,0.19770298
0979575],doc{get,0.17721369744885923],doc{telnet,0.17217377649686889],doc{rm,0.063038323821746946],doc{rmdir,0.063038323821746941
,doc{mkstr,0.062718417048236841,
doc{install,0.059329379705896891
,doc{pack,0.058438553962399211,d
oc{pcat,0.058438553962399211,doc
{unpack,0.058438553962399211,doc
{rmdel,0.055014261247491661,doc{
'sccs-
rmdel',0.055014261247491661],doc{
'sccs-
val',0.053038139001570181,doc{va
l,0.053038139001570181],doc{ftp,0
.052564809363048981,doc{delta,0
.051799427685103115],doc{'sccs-
comb',0.051799427685103115],doc{
size,0.047164410230151841,doc{c
rypt,0.046084812717562801,doc{o
rganizer,0.045372606958813705],d
oc{get,0.044276103621272911,doc{'
sccs-
get',0.044276103621272911,doc{tai
l,0.042765669271421051,doc{what,
0.042119148672803911,doc{prt,0.0
4195381592279415],doc{'sccs-
prt',0.041953815922794151,doc{sp
lit,0.040141207869257911,doc{ch
grp,0.039900991594704831,doc{com
press,0.039747386077868554],doc{
uncompress,0.039747386077868554]
,doc{zcat,0.039747386077868554],
doc{'sccs-
sccsdiff',0.039586937445976561,d
oc{sccsdiff,0.039586937445976561
,doc{admin,0.038415278601618951,
doc{'sccs-
admin',0.038415278601618951],doc{
mv,0.037975546119050481,doc{sum
,0.037975546119050481],doc{tftp,
0.037838761192465331,doc{chmod,0
.037830983987586511,doc{vswap,0.0
3749525328914131,doc{egrep,0.035
780790296222971,doc{fgrep,0.035
780790296222971],doc{grep,0.0357
807902962222971,doc{more,0.03539
565321670632611,doc{page,0.035395
653216706326],doc{'sccs-
unget',0.035349162106084641,doc
{unget,0.035349162106084641],doc
{'old-
ccat',0.035204963919059801,doc{
'old-
compact',0.035204963919059801,d
oc{'old-
uncompact',0.035204963919059801
,doc{find,0.035092399500445771,d
oc{acctcom,0.035041809067252671,
doc{sact,0.034903684141118241,do
c{'sccs-
show',0.034903684141118241,doc{i
ndent,0.034341596585279661,doc{b
ar,0.034312344320276314],doc{tar
,0.034312344320276314],doc{ar,0.
033453724395809285],doc{file,0.0
333612922932846511,doc{screendump
,0.033150343839154085],doc{scree
nload,0.033150343839154085],doc{
cluster,0.032796719592639571,doc
{diff,0.032764087187053411,doc{t
ouch,0.032383414201627681,doc{c
ut,0.032276821614288051,doc{head
,0.032060142274715331,doc{'old-
filemerge',0.032004024640222891,
doc{checknr,0.031815371679478841
,doc{dos2unix,0.031109085111421621
24],doc{unix,0.031109085111421621
4],doc{ln,0.030666347036888083],
doc{make,0.030288698193811295],d
oc{objdump,0.030217913747210311,
doc{diffmk,0.030201518308953391
,doc{strings,0.030012918015922062
],doc{strip,0.029951075797640831
6],doc{df,0.029604069585474681,d
oc{uudecode,0.029366329196730472
},doc{uuencode,0.029366329196730
472],doc{ctags,0.028908098852138
662],doc{csplit,0.028620026960022922],doc{crontab,0.028473544352
106181,doc{paste,0.026911021253636
86093],doc{error,0.026302665935858
1744],doc{du,0.026198657954192261,doc{cmp,0.026170050459737356],
doc{sdiff,0.025350998484127821,d
oc{pr,0.025131974325300564]]}}.
query_doc_rel_bp{reference{17},
docRel{[doc{indent,0.561600414669
4601],doc{sccs,0.413733604475398861],doc{mkstr,0.343765505877219
431,doc{dos2unix,0.335428783669972
7275],doc{unix,0.335428783669972
75],doc{dis,0.2830302271915964],
doc{cc,0.2698219287691792],doc{t
roff,0.2616962291221528],doc{dbx
,0.2487669867054140311,doc{fdform
at,0.2479547546407228],doc{nroff
,0.2454779948084313811,doc{roffbi
b,0.2277347745113581],doc{tbl,0.
213371425615571,doc{lint,0.20904
1097827961681,doc{dd,0.202270717
1137199],doc{foption,0.199250686
61928383],doc{cpp,0.193657300234
9677],doc{whereis,0.184653603561
77732],doc{cb,0.17845110993139032],doc{dbxtool,0.17416483592130866],doc{ctrace,0.164167148489632
62],doc{error,0.16095484655317341
],doc{cxref,0.15987601782736452]
,doc{cflow,0.15770294321966116],
doc{xstr,0.13479735621699984],do
c{csh,0.10967871779184045]]}}.
query_doc_rel_bp{reference{18},
docRel{[doc{tabs,0.51772364386268
87],doc{kill,0.37888785177231034
],doc{ps,0.31342219387875375],do
c{ustar,0.28013630337722983],doc
{wait,0.22121230473777311,doc{acc
tcom,0.20930785125353318],doc{mp
s,0.20149571602327923],doc{gcore
,0.17507096873692543]]}}.
query_doc_rel_bp{reference{19},
docRel{[doc{split,0.331646179275
0387],doc{test,0.2598357626445401],doc{csplit,0.23645832041116593],doc{awk,0.22072785032030273],
doc{nawk,0.22072785032030273],do
c{lptest,0.19938062643919621,doc
{cp,0.05162495940417251,doc{rm,0
.03735808058569342],doc{rmdir,0.
037358080585693421,doc{mkstr,0.0
3716849586484181,doc{rcp,0.03670
847811531678],doc{install,0.0351
60067936125471,doc{pack,0.034632
142415651611,doc{pcat,0.03463214
241565161],doc{unpack,0.03463214
241565161],doc{rmdel,0.032602821
2751617],doc{'sccs-
rmdel',0.032602821275161771,doc{'
sccs-
val',0.031431722019428511,doc{val
,0.031431722019428511,doc{ftp,0.0
3115121508797085611,doc{delta,0.0
3069763084476718711,doc{'sccs-
comb',0.030697630844767187],doc{
uusend,0.028249925099497841,doc{
size,0.027950804071774371,doc{cr
ypt,0.027311007699817191,doc{org
anizer,0.026888936831060786],doc{
get,0.026239121645766292],doc{'s
ccs-
get',0.026239121645766292],doc{u
upick,0.025723190027920027],doc{
uuto,0.025723190027920027],doc{t
ail,0.025344000634606116],doc{rd
ist,0.024963428886680288],doc{wha
t,0.024960856427095991,doc{cpio,
0.024874465398583383],doc{paxcpi
o,0.024874465398583383],doc{prt,
0.024862876122039274],doc{'sccs-
prt',0.024862876122039274],doc{c
hgrp,0.023646321302246041,doc{co
mpress,0.023555290847619691,doc{
uncompress,0.023555290847619691,
doc{zcat,0.023555290847619691],do
c{'sccs-
sccsdiff',0.023460204992592281,d
oc{sccsdiff,0.023460204992592281
,doc{admin,0.022765850782759188
],doc{'sccs-
admin',0.022765850782759188],doc
{mv,0.022505254362613372],doc{su
m,0.022505254362613372],doc{tftp
,0.022424192208665151,doc{chmod,
0.022419583243372603],doc{dd,0.0
222346255195173221,doc{vswap,0.0
222206208705298151,doc{egrep,0.0
21204587404415321,doc{fgrep,0.02
1204587404415321],doc{grep,0.0212
045874041532],doc{more,0.0209766
345579747191,doc{page,0.02097634
557974719],doc{'sccs-
unget',0.020948793789791061,doc{
unget,0.020948793789791061],doc{'
old-
ccat',0.020863338353088498],doc{
'old-
compact',0.020863338353088498],d
oc{'old-
uncompact',0.020863338353088498]
,doc{find,0.020796629875344771},
doc{acctcom,0.020766648724744431
```

```
,doc(sact,0.020684792453693233),doc('sccs-show',0.020684792453692333),doc(indent,0.020351685370025883),doc(bar,0.020334349749298576),doc(tar,0.020334349749298576),doc(ar,0.019825510199227046),doc(file,0.019770732633367507),doc(screendump,0.019645719326047966),doc(cluster,0.019436152791003344),doc(diff,0.019416813737229883),doc(touch,0.019191217234469173),doc(cut,0.019128048301637224),doc(head,0.018999638728884428),doc('old-filemerge',0.018966382021146058),doc(checknr,0.018854581578448146),doc(dos2unix,0.018436018569052706),doc(unix,0.018436018569052706),doc(ln,0.018173640960257965),doc(pg,0.018117926210310455),doc(make,0.017948818721404053),doc(objdump,0.017907888225136687),doc(diffmk,0.017898171880118193),doc(strings,0.017786402649618456),doc(strip,0.017749753410963555),doc(df,0.017544108887887746),doc(uudecode,0.017403218012896977),doc(uuencode,0.017403218012896977),doc(ctags,0.017131659299050497),doc(crontab,0.016874131480306552),doc(paste,0.015948141379541092),doc(error,0.015587614867860804),doc(du,0.015525977147763085),doc(cmp,0.015509023634116003),doc(sdiff,0.015023633035926642),doc(pr,0.014886585426847523)}}}).
query_doc_rel_bp(reference(20),
  docRel[[doc(mail,0.3545998835800116),doc(enroll,0.34610555663166853),doc(xget,0.34610555663166853),doc(xsend,0.34610555663166853),doc(uusend,0.26237511210985204),doc(strings,0.23223622691685362),doc(uudecode,0.22723300304985082),doc(uuencode,0.22723300304985082),doc(lpr,0.22635597730852466),doc(whereis,0.21188121493560338),doc('old-prmail',0.20049288515531338),doc(mailtool,0.18563622774493893),doc(cancel,0.17255665877138411),doc(lp,0.17255665877138411),doc(kill,0.15184917418132984),doc(vacation,0.14583015050270182),doc(fmt,0.14364365754727235),doc(fmt_mail,0.14364365754727235),doc(bin,0.13912520584909027),doc(biff,0.11648709233975585),doc(from,0.10446729860980526),doc(cp,0.062392423584749381),doc(rm,0.045149889028821166),doc(rmdir,0.045149889028821166),doc(mkstr,0.044920762452353331),doc(rcp,0.044364798387371424),doc(install,0.042493434900548105),doc(pack,0.041855399484731178),doc(pcat,0.041855399484731178),doc(unpack,0.041855399484731178),doc(rmdel,0.039402821010128671),doc('sccs-rmdel',0.039402821010128671),doc('sccs-val',0.037987464530108171),doc(val,0.037987464530108171),doc(ftp,0.037648452015852341),doc(delta,0.037100263299387336),doc('sccs-comb',0.037100263299387336),doc(size,0.033780528397655941),doc(crypt,0.033007289121386174),doc(organizer,0.032497186551651985),doc(get,0.031711838829690184),doc('sccs-get',0.031711838829690184),doc(upick,0.031088298890619016),doc(uuto,0.031088298890619016),doc(tail,0.030630021624747241),doc(rdist,0.030170073685467911),doc(what,0.030166964685527976),doc(cpio,0.030062555002555037),doc(paxcpio,0.030062555002555037),doc(prt,0.030048548540186317),doc('sccs-prt',0.030048548540186317),doc(split,0.028750305701412165),doc(chgrp,0.028578255788256851),doc(compress,0.028468239029895961),doc(uncompress,0.028468239029895961),doc(zcat,0.028468239029895961),doc('sccs-sccsdiff',0.028353321032627591),doc(sccsdiff,0.028353321032627591),doc(admin,0.027514144741202581),doc('sccs-admin',0.027514144741202581),doc(mv,0.027199195491497352),doc(sum,0.027199195491497352),doc(tftp,0.027101226131245982),doc(chmod,0.027095655869920091),doc(dd,0.026872121347370576),doc(vswap,0.026855195736157137),doc(egrep,0.025627247256860651),doc(fgrep,0.025627247256860651),doc(grep,0.025627247256860651),doc(more,0.025351400829690403),doc(page,0.025351400829690403),doc('sccs-unget',0.025318102538141411),doc(unget,0.025318102538141411),doc('old-ccat',0.025214823584202885),doc('old-compact',0.025214823584202885),doc('old-uncompact',0.025214823584202885),doc(find,0.025134201659302267),doc(acctcom,0.025097967313175817),doc(sact,0.024999038206102508),doc('sccs-show',0.024999038206102508),doc(indent,0.024596454678617583),doc(bar,0.024575503351894889),doc(tar,0.024575503351894889),doc(ar,0.023960534679548819),doc(file,0.023894332107546296),doc(screendump,0.023743244662364087),doc(screenload,0.023743244662364087),doc(cluster,0.023489968646758444),doc(diff,0.023466596029158585),doc(touch,0.023193947252027728),doc(cut,0.023117602527138015),doc(head,0.022962410454388591),doc('old-filemerge',0.022922217365228127),doc(checknr,0.022787098603716676),doc(dos2unix,0.022281235531270427),doc(unix,0.022281235531270427),doc(ln,0.021964133588798931),doc(pg,0.021896798363381154),doc(make,0.021692419973529508),doc(objdump,0.021642952566869596),doc(diffmk,0.021631209675038086),doc(strip,0.021451835432381581),doc(df,0.021203299446305621),doc(csplit,0.020498499371601711),doc(crontab,0.020393584248687043),doc(paste,0.019274459560376421),doc(error,0.018838737171013288),doc(du,0.018764244155358687),doc(cmp,0.018743754632133161),doc(sdiff,0.018157125680637508),doc(pr,0.017991493985619731)}]).
query_doc_rel_bp(reference(21),
  docRel[[doc(egrep,0.5698419401762231),doc(fgrep,0.5698419401762231),doc(grep,0.5698419401762231),doc(which,0.28221214999844246),doc(apropos,0.27809133573405621),doc(whereis,0.25367806014985281),doc(test,0.17944084995160292),doc(expr,0.16849266484522522),doc(cpio,0.13516346277354457),doc(paxcpio,0.13516346277354457),doc(login,0.13472341505146231),doc(look,0.13380411690697224),doc(gfxtool,0.10214866176697406),doc(shelltool,0.097017929559118449),doc(clock,0.096852518463375141),doc(strings,0.096648178519800014),doc(newgrp,0.094045919733543551),doc(who,0.092260830826529871),doc(lookbib,0.091191993139748671),doc(wall,0.086532069280472141),doc(users,0.085798984478053131),doc(getopt,0.084547480077186971),doc(getoptcvt,0.084547480077186971),doc(getopts,0.084547480077186971),doc(ypcat,0.083481233127494751),doc(yppasswd,0.082864710750979481),doc(pr,0.080891082599165651),doc(banner,0.080611369348102711),doc(rev,0.078901399110910021),doc(comm,0.078099980119653841),doc(inline,0.076279962815819881),doc(rusers,0.076173111373285121),doc(rwho,0.076173111373285121),doc(logname,0.072910556550450521),doc(lastcomm,0.072495748297769431),doc(cp,0.066255597265327864),doc(perfmeter,0.059687758195812321),doc(w,0.057894486433987831),doc(rm,0.047945722267524511),doc(rmdir,0.047945722267524511),doc(mkstr,0.047702407401514005),doc(rcp,0.047112016168584751),doc(install,0.045124777155269554),doc(pack,0.044447226831725271),doc(pcat,0.044447226831725271),doc(unpack,0.044447226831725271),doc(rmdel,0.041842776435234461),doc('sccs-rmdel',0.041842776435234461),doc('sccs-val',0.040339776311633151),doc(val,0.040339776311633151),doc(ftp,0.039979771002360961),doc(delta,0.039397963659372389621),doc('sccs-comb',0.039397963659372389621),doc(uusend,0.036256227345298785),doc(size,0.035872332522685804),doc(crypt,0.035051211665385225),doc(organizer,0.034509521825996831),doc(get,0.033675542727253591),doc('sccs-get',0.033675542727253591),doc(uupick,0.033013391094448211),doc(uuto,0.033013391094448211),doc(tail,0.032526735756336911),doc(rdist,0.032038306291092171),doc(what,0.032035004771401721),doc(prt,0.031909255899396531),doc('sccs-prt',0.031909255899396531),doc(split,0.030530621490263381),doc(chgrp,0.030347917666849092),doc(compress,0.030231088300163887),doc(uncompress,0.030231088300163887),doc(zcat,0.030231088300163887),doc('sccs-sccsdiff',0.030109054193345763),doc(sccsdiff,0.030109054193345763),doc(admin,0.029217913278769631),doc('sccs-admin',0.029217913278769631),doc(mv,0.028883461310458191),doc(sum,0.028883461310458191),doc(tftp,0.028779425357361145),doc(chmod,0.028773510166687735),doc(dd,0.028536136341384),doc(vswap,0.028518159931331666),doc(more,0.026921244497796938),doc(page,0.026921244497796938),doc('sccs-unget',0.026885884744025354),doc(unget,0.026885884744025354),doc('old-ccat',0.026776210409312346),doc('old-compact',0.026776210409312346),doc('old-uncompact',0.026776210409312346),doc(find,0.026690596103206511),doc(acctcom,0.026652118004294341),doc(sact,0.026547062873618757),doc('sccs-show',0.026547062873618757),doc(indent,0.026119550017807504),doc(bar,0.026097301314349095),doc(tar,0.026097301314349095),doc(ar,0.025444251709980378),doc(file,0.025373949651683568),doc(screendump,0.025213506362881864),doc(screenload,0.025213506362881864),doc(cluster,0.024944546643102663),doc(diff,0.024919726714278643),doc(touch,0.024630194606313736),doc(cut,0.024549122358853374),doc(head,0.024384320270116992),doc('old-filemerge',0.024341638289465141),doc(checknr,0.024198152519025608),doc(dos2unix,0.023660964700088009),doc(unix,0.023660964700088009),doc(ln,0.023324226737813876),doc(pg,0.023252721888386174),doc(make,0.023035687700083428),doc(objdump,0.022983157104947371),doc(diffmk,0.022970687053691836),doc(strip,0.022780205353617966),doc(df,0.022516279181966102),doc(uudecode,0.022335458469113157),doc(uuencode,0.022335458469113157),doc(ctags,0.021986937386946126),doc(csplit,0.021767835512164095),doc(crontab,0.021656423691379358),doc(paste,0.020467999303203593),doc(error,0.020005295816877777),doc(du,0.019926189370230308),doc(cmp,0.019904431066690403),doc(sdiff,0.019281476074164181]]}),
query_doc_rel_bp(reference(22),
  docRel[[doc(awk,0.29814996802899
```

```
363),doc(nawk,0.2981499680289936
3),doc(lptest,0.2693150289473864
),doc(cpio,0.14225634041878665),
doc(paxcpio,0.14225634041878665)
,doc(login,0.14179320062295342),
doc(look,0.14082568216804944),do
c(gfxtool,0.10750899858848023),d
oc(shelltool,0.10210907101257512
),doc(clock,0.10193497972174951),
doc(strings,0.10171991677432513)
,doc(newgrp,0.09898110109111113)
,doc(who,0.09710233733333891),doc
(lookbib,0.09597741100559361),do
c(wall,0.09107295161066442),doc(
users,0.09030139723443871),doc(ge
topt,0.08898421852036931),doc(get
optcvt,0.08898421852036931),doc(g
etopts,0.08898421852036931),doc(y
pcat,0.08786201888199484),doc(yp
passwd,0.08721314369583791),doc(
pr,0.08513594685237233),doc(bann
er,0.08484155528643915),doc(rev,
0.08304185214790973),doc(comm,0.
08219837765784539),doc(inline,0.
08028285259042331),doc(rusers,0.
08017039398014841),doc(rwho,0.08
017039398014841),doc(logname,0.0
7673663237040255),doc(lastcomm,0
.07630005651779373),doc(cp,0.069
73284057048762),doc(perfmeter,0.
06281995055841103),doc(w,0.06093
257453993103),doc(rm,0.050461736
09757911),doc(rmdir,0.0504617360
9757911),doc(mkstr,0.05020565296
906284),doc(rcp,0.04958428019206
7884),doc(install,0.047492752345
66457),doc(pack,0.04677965258583
5586),doc(pcat,0.046779652585835
586),doc(unpack,0.04677965258583
5586),doc(rmdel,0.044038530283961
874),doc('sccs-
rmdel',0.044038530283961874),doc
('sccs-
val',0.042456658285711751),doc(va
1,0.042456658285711751),doc(ftp,0
.042077761229791311),doc(delta,0.
041465078564629734),doc('sccs-
comb',0.041465078564629734),doc(
uusend,0.038158819800103941),doc(
size,0.037754779599815951),doc(cr
ypt,0.036890569362787451),doc(org
anizer,0.036320453564799151),doc(g
et,0.035442710416658786),doc('sc
s-
get',0.035442710416658786),doc(u
upick,0.034745811519927851),doc(u
uto,0.034745811519927851),doc(tai
l,0.034233161831309228),doc(rdist
,0.033719557879505771),doc(what,0
.033716083108296561),doc(prt,0.03
358373540148141),doc('sccs-
prt',0.03358373540148141),doc(spl
it,0.032132755367422536),doc(chg
rp,0.031940463924088541),doc(com
press,0.031817503785499766),doc(
uncompress,0.031817503785499766)
,doc(zcat,0.031817503785499766),
doc('sccs-
sccsdiff',0.031689065780982921),d
oc(sccsdiff,0.031689065780982921)
,doc(admin,0.030751161093549407)
,doc('sccs-
admin',0.030751161093549407),doc
(mv,0.030399158325334175),doc(su
m,0.030399158325334175),doc(tftp
,0.030289662950949148),doc(chmod
,0.030283437352989374),doc(dd,0.
030033604178905795),doc(vswap,0.
030014687283544266),doc(egrep,0.
028642271682165307),doc(fgrep,0.
028642271682165307),doc(grep,0.0
28642271682165307),doc(more,0.02
833397215118662 8),doc(page,0.028
333972151186628),doc('sccs-
unget',0.028296756343201171),doc(
unget,0.028296756343201171),doc('
old-
ccat',0.028181326705827365),doc(
'old-
compact',0.028181326705827365),d
oc('old-
uncompact',0.028181326705827365)
,doc(find,0.028091219678201745),
doc(acctcom,0.028050722391248634
),doc(sact,0.027940154356622556)
,doc('sccs-
show',0.027940154356622556),doc(
indent,0.027490207210391281),doc(
bar,0.027466790977423355),doc(ta
r,0.027466790977423355),doc(ar,0
.026779471749851652),doc(file,0.
026705480500568 1),doc(screendum
p,0.026536611774202046),doc(scree
nload,0.026536611774202046),doc(c
luster,0.026253544012842963),doc
(diff,0.026227421626130547),doc(
touch,0.025922695946071345),doc(
cut,0.025837369327497211),doc(hea
d,0.025663919035858195),doc('old
-
filemerge',0.025618997263030206)
,doc(checknr,0.025467988189079457
),doc(dos2unix,0.024902604446917
89),doc(unix,0.024902604446917 89
),doc(ln,0.024548195723414455),
doc(pg,0.024472938564467413),doc
(make,0.024244515226235555),doc(o
bjdump,0.024189228020131664),doc
(diffmk,0.024176103586797046),do
c(strip,0.023975626111297961),doc
(df,0.023697850073882133),doc(uu
decode,0.023507540560093833),doc
(uuencode,0.023507540560093833),
doc(ctags,0.023140703083064384),
doc(csplit,0.022910130853829126)
,doc(crontab,0.022792872553553323
),doc(paste,0.021542083772084303
),doc(error,0.021055099606854068
),doc(du,0.020978419470271251),d
oc(cmp,0.020948941848355575),doc(
sdiff,0.020293296486333115)]}).
```


A.5.3 Sistema MV-S


```
query_doc_rel_th(reference(1),
  docRel([doc('sccs-
  sccsdiff',0.8036215864306511),doc
  (sccsdiff,0.8036215864306511),doc
  (dircmp,0.5896339882187374),doc(
  diff,0.4149092198051469),doc(com
  m,0.26040388694306854),doc(cmp,0
  .24089111888406724),doc(sdiff,0.
  2333518961335278),doc(rastrepl,0
  .2288405128842072),doc(cp,0.0874
  45176072555351),doc(rm,0.063279157
  45120452),doc(rmdir,0.063279157
  45120452),doc(mkstr,0.0629580284
  95620361),doc(rcp,0.06217882530460
  605),doc(install,0.059556043566
  57378),doc(pack,0.05866181448390
  061),doc(pcat,0.05866181448390061
  ),doc(unpack,0.05866181448390061
  ),doc(rmdel,0.05522443949153294
  ),doc('sccs-
  rmdel',0.05522443949153294),doc(
  'sccs-
  val',0.05324076760495028),doc(va
  l,0.05324076760495028),doc(ftp,0
  .05276562964273226 5),doc(delta,0
  .05199732387613294),doc('sccs-
  comb',0.05199732387613294),doc(u
  usend,0.04785126618738668),doc(s
  ize,0.04734459865218338),doc(cry
  pt,0.04626087661071179),doc(orga
  nizer,0.04554594991829247),doc(g
  et,0.04444525746431452),doc('scc
  s-
  get',0.04444525746431452),doc(uu
  pick,0.04357134430903718),doc(uu
  to,0.04357134430903718),doc(tail
  ,0.04292905260274094),doc(rdist,
  0.04228441938890436 4),doc(what,0
  .04228006201614117),doc(cpio,0.0
  4213372817323721),doc(paxcpio,0.
  04213372817323721),doc(prt,0.042
  11409762360289),doc('sccs-
  prt',0.04211409762360289),doc(sp
  lit,0.04029456462425825 4),doc(ch
  grp,0.04005343061976332 5),doc(co
  mpress,0.03989923826349539),doc(
  uncompress,0.03989923826349539),
  doc(zcat,0.03989923826349539),do
  c(admin,0.03856204156121814),doc
  ('sccs-
  admin',0.03856204156121814),doc(
  mv,0.03812062911060251),doc(sum,
  0.03812062911060251),doc(tftp,0.
  03798331216050915),doc(chmod,0.
  03797551468934572),doc(dd,0.0376
  62223194007576),doc(vswap,0.0376
  38501354573115),doc(egrep,0.0359
  17488372399999),doc(fgrep,0.03591
  748837239999),doc(grep,0.0359174
  88372399999),doc(more,0.035530879
  902861655),doc(page,0.0355308799
  02861655),doc('sccs-
  unget',0.035484211176113306),doc
  (unget,0.035484211176113306),doc
  ('old-
  ccat',0.03533946208972064),doc('
  old-
  compact',0.03533946208972064),do
  c('old-
  uncompact',0.03533946208972064),
  doc(find,0.035226467626399841),do
  c(acctcom,0.03517568391589695),d
  oc(sact,0.035037031292874936),do
  c('sccs-
  show',0.035037031292874936),doc(
  indent,0.034472796319745371),doc(
  bar,0.03444343229845951),doc(tar,
  0.03444343229845951),doc(ar,0.033
  58153207495854),doc(file,0.03348
  87468418717251),doc(screendump,0.
  03327699247351697),doc(screenloa
  d,0.03327699247351697),doc(clust
  er,0.032922017229615644),doc(tou
  ch,0.032507132833462811),doc(cut,
  0.03240013301631917),doc(head,0.0
  3218262586813667),doc('old-
  filemerge',0.03212629384003774),
  doc(checknr,0.03193692014348586)
  ,doc(dos2unix,0.03122793525562395
  8),doc(unix,0.03122793525562395
  8),doc(ln,0.030783505730383538),
  doc(pg,0.030689133043698094),doc
  (make,0.03040268954208031),doc(ob
  jdump,0.030333592318167021),doc(
  diffmk,0.030316901155902843),doc
  (strings,0.030127580328277094),d
  oc(strip,0.030065501846006291),do
  c(df,0.029717169920203512),doc(u
  udecode,0.029478521260470831),doc
  (uuencode,0.029478521260470831),d
  oc(ctags,0.029018540278007873),d
  oc(csplit,0.028729367826817637),
  doc(crontab,0.028582325592058398
  ),doc(paste,0.027813832980394781
  ),doc(error,0.026403153482403311),
  doc(du,0.026298748145357032),doc
  (pr,0.025215711523922393)]]}).
query_doc_rel_th(reference(2),
  docRel([doc(ld,0.47740537667257
  466),doc('ld.so',0.47740537667257
  466),doc(paste,0.44520731837634
  92),doc(ypmatch,0.41822720717913
  797),doc(cu,0.34841774787234211),
  doc(tip,0.34841774787234211),doc(f
  ind,0.30886376143641661),doc(line
  ,0.29023844144386435),doc(talk,0
  .27846321279066955),doc(write,0.
  27843812459396494),doc(ln,0.2519
  04358722582931),doc(diff,0.236252
  97299176054),doc('sccs-
  sccsdiff',0.21321381927821971),do
  c(sccsdiff,0.21321381927821971),d
  oc(adjacentscreens,0.21261649199
  260715),doc(toolplaces,0.2052963
  078087293),doc(whatis,0.19973348
  58100958),doc(pr,0.19877217010798368),doc(colrm,0.18831995930175335),doc(yacc,0.18458397782990954),doc(cmp,0.14095094920739921),doc(comm,0.13738588960617615),doc
  (sdiff,0.13653956476662507),doc(
  rastrepl,0.12073344146127872),do
  c(cp,0.09226999986465836),doc(rm
  ,0.06677061116114365),doc(rmdir,
  0.06677061116114365),doc(mkstr,0
  .06643176378248776),doc(rcp,0.06
  5609567732819 15),doc(install,0.0
  6284207292655941),doc(pack,0.0618
  985043840389261),doc(pcat,0.06189
  850438403892 6),doc(unpack,0.0618
  985043840389261),doc(rmdel,0.0582
```

```
71470803394246),doc('sccs-
rmdel',0.058271470803394246),doc
('sccs-
val',0.056178348999229291),doc(va
l,0.056178348999229291),doc(ftp,0
.055676995103238 8),doc(delta,0.0
5486629774788 88),doc('sccs-
comb',0.05486629774788 88),doc(uu
send,0.05049147960969823),doc(si
ze,0.049956856483479 74),doc(cryp
t,0.048813339629709 07),doc(organ
izer,0.048058966561055 9),doc(get
,0.046897542945243 07),doc('sccs-
get',0.046897542945243 07),doc(uu
pick,0.045975411269823 29),doc(uu
to,0.045975411269823 29),doc(tail
,0.045297680852723406),doc(rdist
,0.044617479734435 28),doc(what,0
.044612881941825 92),doc(cpio,0.0
444584740685575 26),doc(paxcpio,0
.044458474068557526),doc(prt,0.0
444377603952200 86),doc('sccs-
prt',0.044437760395220086),doc(s
plit,0.042517833909350 02),doc(ch
grp,0.042263395236327 61),doc(com
press,0.042100695252966 96),doc(u
ncompress,0.042100695252966 96),d
oc(zcat,0.042100695252966 96),doc
(admin,0.040689718169042176),doc
('sccs-
admin',0.040689718169042176),doc
(mv,0.040223950603719 09),doc(sum
,0.040223950603719 09),doc(tftp,0
.040079067101239 72),doc(chmod,0.
040070829434192 09),doc(dd,0.0397
402513 50896),doc(vswap,0.0397152
212491754 3),doc(egrep,0.03789925
066318606 4),doc(fgrep,0.03789925
066318606 4),doc(grep,0.037899250
663186064),doc(more,0.037491310 9
11285054),doc(page,0.037491310 91
1285054),doc('sccs-
unget',0.037442067218217 51),doc(
unget,0.037442067218217 51),doc('
old-
ccat',0.037289331541057014),doc(
'old-
compact',0.037289331541057014),d
oc('old-
uncompact',0.037289331541057014)
,doc(acctcom,0.037116516838688 45
),doc(sact,0.036970213999789 01),
doc('sccs-
show',0.036970213999789 01),doc(i
ndent,0.036374847128423 76),doc(b
ar,0.036343862935106 74),doc(tar,
0.036343862935106 74),doc(ar,0.03
543440700994745),doc(file,0.0353
365023112198),doc(screendump,0.
035113064309916 96),doc(screenloa
d,0.035113064309916 96),doc(clust
er,0.034738503159973635),doc(tou
ch,0.034300727345500 94),doc(cut,
0.034187823769225016),doc(head,0
.033958315574017066),doc('old-
filemerge',0.033898875402946324)
,doc(checknr,0.033699052934286 49
6),doc(dos2unix,0.032950949511171
7086),doc(unix,0.032950949511717
086),doc(pg,0.032382418666848116
),doc(make,0.032080170524276 96),
doc(objdump,0.032007014885441 83)
,doc(diffmk,0.031989648728375 78)
,doc(strings,0.031789882045706 96
4),doc(strip,0.031724378357476 37
),doc(df,0.031356827073458 03),do
c(uudecode,0.031105010875124052)
,doc(uuencode,0.031105010875124 0
52),doc(ctags,0.030619650251521 5
93),doc(csplit,0.030314522590619
553),doc(crontab,0.030159367239 6
81992),doc(error,0.027859958406 6
9951),doc(du,0.027747972461957 9)
]}]).
query_doc_rel_th(reference(3),
docRel([doc(rm,0.6874643643773599
),doc(rmdir,0.6874643643773599)
,doc(rmdel,0.5999579596554822),d
oc('sccs-
rmdel',0.5999579596554822),doc(k
eylogout,0.5822580990530402),doc
(cancel,0.5683182053469306),doc(
lp,0.5683182053469306),doc(overv
iew,0.4755095324384396),doc(lpr
m,0.3967820753853405),doc(cut,0.
351994839170091 5),doc(unwhiteout
,0.3411725309456539),doc(strip,0
.3266314209728407 6),doc(crontab,
0.3105182035763206 5),doc(uniq,0.
309886086212240 46),doc(atrm,0.29
675577924570823),doc(colrm,0.270
010095541627 5),doc(deroff,0.2688
116291692854),doc(unifdef,0.2443
213138591376 3),doc(iperm,0.24273
077004510052),doc(cp,0.093964781
955199 71),doc(mkstr,0.0676519584
6837459),doc(rcp,0.066814660618 0
9974),doc(install,0.063996333465
02961),doc(pack,0.063035433795713
77),doc(pcat,0.063035433795713 7
7),doc(unpack,0.063035433795713 7
7),doc('sccs-
val',0.0572102126591375),doc(val
,0.0572102126591375),doc(ftp,0.0
5669965007554 33),doc(delta,0.055
8740621234585 8),doc('sccs-
comb',0.055874062123458 58),doc(u
usend,0.051418688890300014),doc(s
ize,0.050874446011951985),doc(cr
ypt,0.049709925452894 16),doc(org
anizer,0.048941696331697204),doc
(get,0.047758940105648 13),doc('s
ccs-
get',0.047758940105648 13),doc(uu
pick,0.046819871048079 07),doc(uu
to,0.046819871048079 07),doc(tail
,0.046129692323026425),doc(rdist
,0.045436997515836 9),doc(what,0.
045432315272630586),doc(cpio,0.0
452750712912161 2),doc(paxcpio,0.
045275071291216 12),doc(prt,0.045
253977156594 91),doc('sccs-
prt',0.045253977156594 91),doc(sp
lit,0.043298786153241 57),doc(chg
rp,0.043039674042408 26),doc(comp
ress,0.042873985644413),doc(unco
mpress,0.042873985644413),doc(zc
at,0.042873985644413),doc('sccs-
sccsdiff',0.042700915839849585),
doc(sccsdiff,0.042700915839849 58
5),doc(admin,0.041437092242123 55
),doc('sccs-
admin',0.041437092242123 55),doc(
mv,0.040962769626088 18),doc(sum,
0.040962769626088 18),doc(tftp,0.
040815224955671 49),doc(chmod,0.0
408068359821255 26),doc(dd,0.0404
701865708024 7),doc(vswap,0.04044
469611417725),doc(egrep,0.038595
370435193 43),doc(fgrep,0.038595
370435193 43),doc(grep,0.038595370
435193 43),doc(more,0.038179937793
00786),doc(page,0.038179937793 0
0786),doc('sccs-
unget',0.038129789609540346),doc
(unget,0.038129789609540346),doc
('old-
ccat',0.037974248538528994),doc(
'old-
compact',0.037974248538528994),d
oc('old-
uncompact',0.037974248538528994)
,doc(find,0.037852829604003 97),d
oc(acctcom,0.037798259638013844)
,doc(sact,0.037649269561317 44),d
oc('sccs-
show',0.037649269561317 44),doc(i
ndent,0.037042967205912186),doc(
bar,0.037011413906103345),doc(ta
r,0.037011413906103345),doc(ar,0
.036085253422405435),doc(file,0.
035985550447498 84),doc(screendum
p,0.035758008415517 5),doc(screen
load,0.035758008415517 5),doc(clu
ster,0.035376567461416084),doc(d
iff,0.035341367628 0186),doc(tou
ch,0.034930750738618 6),doc(head,
0.034582049671169 04),doc('old-
filemerge',0.034521517724466 6),d
oc(checknr,0.034318024988746 3),d
oc(dos2unix,0.033556180672231 92)
,doc(unix,0.033556180672231 92),d
oc(ln,0.033078616039060 85),doc(p
g,0.032977207255784 35),doc(make
,0.032269407530300 65),doc(objdum
p,0.032594908195066 88),doc(diffm
k,0.032577223062689 02),doc(strin
gs,0.032373787137616 71),doc(df,0
.031932779796777 09),doc(uudecode,
0.031676336500295596),doc(uuenco
de,0.031676336500295596),doc(cta
gs,0.031182060947750445),doc(csp
lit,0.030871328811983 51),doc(pas
te,0.029027889699379 75),doc(erro
r,0.028371680078100 38),doc(du,0.
028259490645004926),doc(cmp,0.02
822863283452704),doc(sdiff,0.027
345152784404 6),doc(pr,0.02709570
660858)]}]).
query_doc_rel_th(reference(4),
docRel([doc(chmod,0.659722649714
8246),doc(touch,0.57508250157733
73),doc(mesg,0.39305373007490 07)
,doc(mv,0.3055436048591768 6),doc
(pax,0.25511483006919 82),doc(loc
kscreen,0.25134045314930825),doc
(stty,0.23864609634852492),doc(d
d,0.21369159497556559),doc(vswap
,0.21355699968964154),doc(chgrp,
0.20128402152815544),doc(cd,0.19
030129185001 62),doc(env,0.186399
157232846 42),doc(coloredit,0.178
061299672051 28),doc(dos2unix,0.1
771841045652807 5),doc(unix,0.177
184104565280 75),doc(ranlib,0.168
519719989757 4),doc(chkey,0.16749
025236391 96),doc(cdc,0.159572096
414878 02),doc('sccs-
cdc',0.159572096414878 02),doc(ch
fn,0.157641676883922 42),doc(chsh
,0.157641676883922 42),doc(passwd
,0.157641676883922 42),doc(yppass
wd,0.144666171967012 54),doc(sv_a
cquire,0.129659745917379),doc(s
v_release,0.129659745917379),do
c('old-
sunlcvt',0.12225958854478 74),doc
(rasfilter,0.109805487549717 54),
doc(cp,0.065379668370506 42),doc(
rm,0.047311590126962 55),doc(rmdi
r,0.047311590126962 55),doc(mkstr
,0.047071493531610085),doc(rcp,0
.046488910659974854),doc(install
,0.044527949427550 88),doc(pack,0
.043859365939035 45),doc(pcat,0.0
438593659390354 5),doc(unpack,0.0
438593659390354 5),doc(rmdel,0.04
128936211309996),doc('sccs-
rmdel',0.041289362113099 96),doc(
'sccs-
val',0.039806240734362466),doc(v
al,0.039806240734362466),doc(ftp
,0.039450996868848841),doc(delta,
0.038876561794039 79),doc('sccs-
comb',0.038876561794039 79),doc(u
usend,0.035776700956544 22),doc(s
ize,0.035397883543011 78),doc(cry
pt,0.034587622864727 69),doc(orga
nizer,0.034053097437952685),doc(
get,0.033230148581868416),doc('s
ccs-
get',0.033230148581868416),doc(u
pick,0.032576754594425945),doc(
uuto,0.032576754594425945),doc(t
ail,0.032096535780297 96),doc(rdi
st,0.031614566303716 67),doc(what
,0.031611308450064594),doc(cpio,
0.031501899806270 4),doc(paxcpio,
0.031501899806270 4),doc(prt,0.03
148722273793265 5),doc('sccs-
prt',0.031487222737932655),doc(s
plit,0.030126822205516317),doc(c
ompress,0.029831250653995278),do
c(uncompress,0.029831250653995 27
8),doc(zcat,0.029831250653995278
),doc('sccs-
sccsdiff',0.029710830575409884),
doc(sccsdiff,0.029710830575409 88
4),doc(admin,0.028831475928074 63
7),doc('sccs-
admin',0.028831475928074637),doc
(sum,0.028501447415571 89),doc(tf
tp,0.028398787466036898),doc(egre
p,0.026854236951168 28),doc(fgrep
,0.026854236951168 28),doc(grep,0
.026854236951168 28),doc(more,0.0
265651834588787 43),doc(page,0.02
656518345887874 3),doc('sccs-
unget',0.026530290900877038),doc
(unget,0.026530290900877038),doc
('old-
ccat',0.026422067123530714),doc(
'old-
compact',0.026422067123530714),d
oc('old-
uncompact',0.026422067123530714)
,doc(find,0.026337585155840608),
doc(acctcom,0.026299615969884053
),doc(sact,0.026195950302788815),
doc('sccs-
show',0.026195950302788815),doc(i
ndent,0.025774091750000 23),doc(b
ar,0.025752137308830126),doc(tar
,0.025752137308830126),doc(ar,0.
025107724966013146),doc(file,0.0
250383527256981 13),doc(screendum
p,0.024880031466586544),doc(scre
```

```
enload,0.0248800314665865441),doc
(cluster,0.024614629019380628),d
oc(diff,0.024590137359979811),doc
(cut,0.024224434633356711),doc(he
ad,0.024061812223981842),doc('ol
d-
filemerge',0.024019694757002413)
,doc(checknr,0.023878106735401482
),doc(ln,0.023015739533352245),
doc(pg,0.022945180410072437),doc
(make,0.022731016725078317),doc(
objdump,0.022679180901804347),do
c(diffmk,0.022666875779971528),d
oc(strings,0.022252327286598597)
,doc(strip,0.022478913398879485)
,doc(df,0.022218477925881525),do
c(uudecode,0.022040048755386831),
doc(uuencode,0.022040048755386831
),doc(ctags,0.021696137227720454
),doc(csplit,0.021479933203555446
),doc(crontab,0.021369994920201
307),doc(paste,0.020197288415418
575),doc(error,0.019740704932457
526),doc(du,0.019662644750995568
),doc(cmp,0.019641174223701142),
doc(sdiff,0.019026458460123861),d
oc(pr,0.018852896537054541)]}).
query_doc_rel_th(reference(5),
  docRel([doc(cd,0.6716296716752251
),doc(mv,0.5923954485666593),do
c(pax,0.5455085449820894),doc(sty
,0.5099045092528259),doc(env,0
.39857476345219561),doc(dd,0.3967
2241781003875),doc(vswap,0.39647
25391600912),doc(coloredit,0.380
7460368939491),doc(rm,0.376533531
14365131),doc(rmdir,0.37653353114
365131),doc(chgrp,0.3663685997554
177),doc(mkdir,0.361141876677690
15),doc(ranlib,0.360343407819398
66),doc(chkey,0.3581421113031411
6),doc(chmod,0.34736190949051593
),doc(cdc,0.34121082694961561),do
c('sccs-
cdc',0.34121082694961561),doc(chf
n,0.33708303732150591),doc(chsh,0
.33708303732150591),doc(passwd,0.
33708303732150591),doc(dos2unix,0
.32894558332386675),doc(unix,0.3
2894558332386675),doc(dircmp,0.3
1742510900101612),doc(yppasswd,0.
30933769297711194),doc(ls,0.2900
1360193961534),doc(sv_acquire,0.
27724965794446155),doc(sv_releas
e,0.27724965794446355),doc(organ
izer,0.2710146285568175),doc('ol
d-
sun3cvt',0.2614260027105664),doc
(rasfilter,0.23479556922679842),
doc(pwd,0.21939867926130358),doc
(du,0.15648692085485952),doc(who
is,0.15060863287219659)]]}).
query_doc_rel_th(reference(6),
  docRel([doc(mail,0.8706917011169
463),doc(enroll,0.849834559524491
5),doc(xget,0.8498345595244915)
,doc(xsend,0.8498345595244915),d
oc(rm,0.6874643643773599),doc(rm
dir,0.6874643643773599),doc(rmde
l,0.5999579596554822),doc('sccs-
rmdel',0.5999579596554822),doc(u
usend,0.5916670515040708),doc(ke
ylogout,0.5822580990530402),doc(
write,0.5671047513571712),doc(lp
r,0.5358049045803835),doc('old-
prmail',0.5129858065659805),doc(
wall,0.5126951146909788),doc(ove
rview,0.4755095324384399 6),doc(m
ailtool,0.4749732138566108),doc(
rwall,0.4441897895254732),doc(ca
ncel,0.4084570912907335 4),doc(lp
,0.4084570912907335 4),doc(lprm,0
.3967820753853405),doc(vacation,
0.3731244493754261),doc(fmt,0.367
53003712771491),doc(fmt_mail,0.36
75300371277149),doc(kill,0.35944
06176071119),doc(bin,0.3559690204
5111155),doc(cut,0.3519948391700
915),doc(unwhiteout,0.3411725309
456539),doc(strip,0.3266314209728
84076),doc(crontab,0.3105182035 7
632065),doc(uniq,0.3098806862122
4046),doc(biff,0.2980466113405714
),doc(atrm,0.29675577924570823)
,doc(colrm,0.2700100955416275),d
oc(deroff,0.2688116291692854),do
c(from,0.2672924846964319 6),doc(
unifdef,0.2443213138591376 3),doc
(ipcrm,0.2427307700451005 2),doc(
cp,0.0939647819551997 1),doc(mkst
r,0.0676519584683745 9),doc(rcp,0
.0668146606180997 4),doc(install,
0.0619963334650296 1),doc(pack,0.
0630354337957137 7),doc(pcat,0.06
303543379571377),doc(unpack,0.06
303543379571377),doc('sccs-
val',0.0572102126591375),doc(val
,0.0572102126591375),doc(ftp,0.0
5669965007554 33),doc(delta,0.055
87406212345858),doc('sccs-
comb',0.0558740621234585 8),doc(s
ize,0.0508744460119519 85),doc(cr
ypt,0.0497099254528941 6),doc(org
anizer,0.0489416963316972 04),doc
(get,0.0477589401056481 3),doc('s
ccs-
get',0.0477589401056481 3),doc(uu
pick,0.0468198710480790 7),doc(uu
to,0.0468198710480790 7),doc(tail
,0.0461296923230264 25),doc(rdist
,0.0454369975158369),doc(what,0.
0454323152726305 86),doc(cpio,0.0
4527507129121612),doc(paxcpio,0.
0452750712912161 2),doc(prt,0.045
253977156594 91),doc('sccs-
prt',0.0452539771565949 1),doc(sp
lit,0.0432978615324157),doc(chg
rp,0.0430396740424082 6),doc(comp
ress,0.0428739856444 13),doc(unco
mpress,0.0428739856444 13),doc(zc
at,0.0428739856444 13),doc('sccs-
sccsdiff',0.0427009158398495 85),
doc(sccsdiff,0.0427009158398495 8
5),doc(admin,0.0414370922421235 5
),doc('sccs-
admin',0.0414370922421235 5),doc(
mv,0.0409627696260881 8),doc(sum,
0.0409627696260881 8),doc(tftp,0.
0408152249556714 9),doc(chmod,0.0
4080635982125526),doc(dd,0.0404
7018657080247),doc(vswap,0.04044
469611417725),doc(egrep,0.038595
37043519343),doc(fgrep,0.0385953
7043519343),doc(grep,0.038595370
43519343),doc(more,0.038179937793
00786),doc(page,0.038179937793 0
0786),doc('sccs-
unget',0.038129789609540346),doc
(unget,0.038129789609540346),doc
('old-
ccat',0.037974248538528994),doc(
'old-
compact',0.037974248538528994),d
oc('old-
uncompact',0.037974248538528994)
,doc(find,0.037852829604003 97),d
oc(acctcom,0.037798259638013844)
,doc(sact,0.037649269561317 44),d
oc('sccs-
show',0.037649269561317 44),doc(i
ndent,0.037042967205912186),doc(
bar,0.037011413906103345),doc(ta
r,0.037011413906103345),doc(ar,0
.036085253422405435),doc(file,0.
035985550447498 84),doc(screendum
p,0.0357580084155175),doc(screen
load,0.0357580084155175),doc(clu
ster,0.035376567461416084),doc(d
iff,0.035341367628001 86),doc(tou
ch,0.034930750738618 6),doc(head,
0.034582049671169 04),doc('old-
filemerge',0.034521517724466 6),d
oc(checknr,0.034318024988746 3),d
oc(dos2unix,0.033556180672231 92)
,doc(unix,0.033556180672231 92),d
oc(ln,0.033078616039060 85),doc(p
g,0.032977207255578435),doc(make
,0.032669407530300 65),doc(objdum
p,0.032594908195066 88),doc(diffm
k,0.032577223062689 02),doc(strin
gs,0.032373787137616 71),doc(df,0
.031932777967770 9),doc(uudecode,
0.031676336500295596),doc(uuenco
de,0.031676336500295596),doc(cta
gs,0.031182060947750445),doc(csp
lit,0.030871328811983 51),doc(pas
te,0.029027889699379 75),doc(erro
r,0.028371680078100 38),doc(du,0.
028259490645004926),doc(cmp,0.02
822863283452704),doc(sdiff,0.027
34515278440 46),doc(pr,0.02709570
66085)]]}).
query_doc_rel_th(reference(7),
  docRel([doc(screendump,0.3785440
826776852),doc(screenload,0.3785
440826776852),doc(date,0.3672966
283678383),doc(rm,0.337511163578
0966),doc(rmdir,0.337511163578090
66),doc(talk,0.31449243134775307
),doc(write,0.31446409709167417)
,doc(stty,0.30691118630062272),do
c(hostname,0.283345597874611),do
c(mkdir,0.2795112179387 98),doc(s
win,0.2688430591251486),doc(stty
_from_defaults,0.25775813565576 09
),doc(domainname,0.2556951669129
96),doc(tabs,0.2549399688224953)
,doc(printenv,0.24617301162864927
),doc(dircmp,0.2456759643605930
2),doc(organizer,0.2429278007540
8535),doc(ls,0.2244604138481161)
,doc(yacc,0.20846654534294404),d
oc(cd,0.20487710344832769),doc(i
pcrm,0.18973991176820013),doc(pw
d,0.16980692634884 08),doc(du,0.1
4026926787119864),doc(cpio,0.129
92451507416616),doc(paxcpio,0.12
992451507416616),doc(login,0.129
50152364047657),doc(look,0.12861
787679768105),doc(whois,0.116565
82945592675),doc(gfxtool,0.09818
932826893431),doc(shelltool,0.09
3257506111134394),doc(clock,0.093
09850633338983),doc(strings,0.09
2902086623222 41),doc(newgrp,0.09
0400691617055 76),doc(who,0.08868
479291300585),doc(lookbib,0.0876
57383468926 24),doc(wall,0.083178
07866809104),doc(users,0.0824734
0829705964),doc(getopt,0.0812704
1231679133),doc(getoptcvt,0.0812
7041231679133),doc(getopts,0.081
2704123167913 3),doc(ypcat,0.0802
4549319260338),doc(yppasswd,0.07
9652867277600 57),doc(pr,0.077755
73712645352),doc(banner,0.077486
86558559 15),doc(rev,0.0758431739
4511672),doc(comm,0.075072818024
41457),doc(inline,0.073323344751
2756),doc(rusers,0.0732206348800
469),doc(rwho,0.0732206348800469
),doc(logname,0.07008453696895792
),doc(lastcomm,0.06968580672061
672),doc(cp,0.06368788512864764)
,doc(perfmeter,0.05734255440967
16),doc(w,0.05565049105027825),d
oc(mkstr,0.04585345789709 2983),d
oc(rcp,0.04528595028025360 3),doc
(install,0.0433757313442 85),doc(p
ack,0.04272444841235353),doc(pca
t,0.04272444841235353),doc(unpac
k,0.04272444841235353),doc(rmdel
,0.04022094674218901),doc('sccs-
rmdel',0.04022094674218901),doc(
'sccs-
val',0.03877620307430 33),doc(va
l,0.03877620307430 33),doc(ftp,0
.03843015156367127),doc(delta,0.
03787058073588656),doc('sccs-
comb',0.03787058073588656),doc(
uusend,0.03485093278609508 6),doc
(size,0.03448191776055494 6),doc(
crypt,0.03369262361968840 5),doc(
get,0.03237027573071437 5),doc('s
ccs-
get',0.03237027573071437 5),doc(u
pick,0.03173378915340656),doc(u
uto,0.03173378915340656),doc(tai
l,0.03126599661898252),doc(rdist
,0.03079649872274869 6),doc(what,
0.03079332517025169),doc(prt,0.0
30672450335579 88),doc('sccs-
prt',0.03067245033557988),doc(sp
lit,0.02934725191734123),doc(chg
rp,0.02917162971018405 4),doc(com
press,0.02905932866002207 4),doc(
uncompress,0.02905932866002207 4)
,doc(zcat,0.02905932866002207 4),
doc('sccs-
sccsdiff',0.02894202460591229 6),
doc(sccsdiff,0.02894202460591229
6),doc(admin,0.02808542439152569
),doc('sccs-
admin',0.02808542439152569],doc(
mv,0.02776393581090782 4),doc(sum
,0.02776393581090782 4),doc(tftp,
0.02766393230049866),doc(chmod,0
.02765824638313576),doc(dd,0.027
43007058515943),doc(vswap,0.0274
12793543535003),doc(egrep,0.02615
93490172224 55),doc(fgrep,0.02615
93490172224 55),doc(grep,0.026159
3490172224 55),doc(more,0.0258777
751559655 24),doc(page,0.02587777
5155965524),doc('sccs-
unget',0.02584378548779 77),doc(u
nget,0.02584378548779 77),doc('ol
```

```
d-
ccat',0.025738362139939682),doc(
'old-
compact',0.025738362139939682),d
oc('old-
uncompact',0.025738362139939682)
,doc(find,0.025656066251864895),
doc(acctcom,0.025619079567449195
),doc(sact,0.025518096382874035)
,doc('sccs-
show',0.025518096382874035),doc(
indent,0.025107153962936855),doc
(bar,0.025085767621180383),doc(t
ar,0.025085767621180383),doc(ar,
0.024458030276886863),doc(file,0
.024390453132549915),doc(cluster
,0.023977693822330608),doc(diff,
0.023953835916129738),doc(touch,
0.023675526098122436),doc(cut,0.
023597596218100073),doc(head,0.02
3439181872811159),doc('old-
filemerge',0.023398154249481995)
,doc(checknr,0.023260230000118592
),doc(dos2unix,0.022743863629028
8),doc(unix,0.022743863629028 8)
,doc(ln,0.022420177659013805),do
c(pg,0.022351444343835 94),doc(ma
ke,0.022142822419751372),doc(obj
dump,0.022092327915100782),doc(d
iffmk,0.022080341204128152),doc(
strip,0.021897242591494472),doc(
df,0.021643546221457692),doc(uud
ecode,0.021469734135330956),doc(
uuencode,0.021469734135330956),d
oc(ctags,0.021134721760947735),do
c(csplit,0.020924112294093203),d
oc(crontab,0.020817018805276345)
,doc(paste,0.019674657590203 95),
doc(error,0.019229888792341776),
doc(cmp,0.019132933568738728),do
c(sdiff,0.018534124366487 05)]]).
query_doc_rel_th(reference(8),
  docRel([doc(mkdir,1.103911539707
  9313),doc(delta,0.663087897960613
  4),doc('sccs-
  comb',0.6630878979606134),doc(pw
  d,0.6303510153240595),doc(hostna
  me,0.5996231094301863),doc(uname
  ,0.5583885057227 88),doc(domainna
  me,0.5411089924294168),doc(ls,0.
  5373829387980056),doc('old-
  syslog',0.5174982390395686),doc(
  swin,0.4703019841790780 4),doc(fi
  nd,0.4604882374069963),doc(nm,0
  .4573339722496981 3),doc(graph,0.
  4410826626754228),doc(script,0.4
  2290835918111 85),doc(vwidth,0.40
  438194455613 33),doc(logname,0.39
  831152373665 12),doc(ln,0.3925619
  355958397),doc(sed,0.38743932488
  273264),doc(look,0.3526202694772
  866),doc(tty,0.3496998522091394)
  ,doc(id,0.34711625803859725),doc
  (rm,0.3440018501874 65),doc(rmdir
  ,0.3440018501874 65),doc(enroll,0
  .3331387604033909 7),doc(xget,0.3
  3313876040339097),doc(xsend,0.33
  3138760403390 97),doc(clear_color
  map,0.3303178473611 5577),doc(env
  ,0.3264971504395436 4),doc(dircmp
  ,0.2900002676090865 4),doc(whois,
  0.2830594425548381 7),doc(overvie
  w,0.2790713262071995 4),doc(date,
  0.2758488692231244 6),doc(lookbib
  ,0.2726205603478406 6),doc('sccs-
  unget',0.2710822065424857 7),doc(
  unget,0.2710822065424857 7),doc(f
  ile,0.2677872367057406 7),doc(lea
  ve,0.2666605618844747),doc(from,
  0.2665114461376091),doc(basename
  ,0.2639251721627102),doc(dirname
  ,0.2639251721627102),doc(trace,0
  .2590065732142213),doc(organizer
  ,0.2475995520724176 8),doc(ps,0.2
  4755459134426722),doc(uptime,0.2
  4424464928984 89),doc(gprof,0.242
  8988070390090 7),doc(cd,0.2418405
  6825267417),doc(cluster,0.241159
  0905040136 6),doc(chkey,0.2397360
  3175514865),doc(which,0.23452927
  830330453),doc(stty,0.2304979059
  2733063),doc(inline,0.2300586396
  6558238),doc(gcore,0.22476931527
  246957),doc(strings,0.2206893891
  2157726),doc(ldd,0.2135073667771
  092),doc(symorder,0.213507366777
  1092),doc(lorder,0.2095571508860
  435),doc(keylogin,0.209206337772
  80148),doc(disablenumlock,0.2085
  2119027326763),doc(enablenumlock
  ,0.2085211902732 6763),doc(mach,0
  .2056544650727 7457),doc(keylogou
  t,0.2009049598477 6743),doc(users
  ,0.1957220771220362),doc(stty_fr
  om_defaults,0.1935827469194604 5)
  ,doc(lsw,0.1917682902139474 2),do
  c(tabs,0.1914662317957 4107),doc(
  xargs,0.1899326191475673 7),doc(a
  rch,0.1864901648507812 6),doc(pri
  ntenv,0.1848820317049229 3),doc(f
  oction,0.1794191062830886),doc(c
  omm,0.1781593386588209),doc(ypma
  tch,0.1660077362544734 6),doc(who
  ami,0.1645643813877 2866),doc(cro
  ntab,0.1597609123520 2067),doc(mp
  s,0.1591501514952057 5),doc(defau
  lts_from_input,0.157611544539990
  96),doc(input,0.157611544539990 9
  6),doc(hostid,0.1443687091350290
  6),doc(clear_functions,0.1442923
  2293635905),doc(du,0.14296679007
  7651),doc(ipcrm,0.142499375342309
  66),doc(toolplaces,0.125099317 6
  2863366)]}).
query_doc_rel_th(reference(9),
  docRel([doc(mkdir,0.697862837495
  8504),doc(newgrp,0.444351924977 3
  744),doc(rm,0.3579538865411791),
  doc(rmdir,0.3579538865411791),do
  c(delta,0.303747508519779),doc('
  sccs-
  comb',0.303747508519779),doc(dir
  cmp,0.301762106314500 1),doc(su,0
  .298930132375752 67),doc(ls,0.275
  7031907590897),doc(organizer,0.2
  5764170140910053),doc(cd,0.25164
  9144567672),doc('old-
  syslog',0.237055752721607 98),doc
  (chkey,0.210332817824833 32),doc(
  pwd,0.2085726859572128 7),doc(adm
  in,0.2071525070998884),doc('sccs
  -
  admin',0.2071525070998884),doc(a
  ddbib,0.1961527513595 8458),doc(s
  cript,0.1937259914216631),doc(vw
  idth,0.1852394057061 6815),doc(ba
  r,0.1850276350753917),doc(tar,0.
  1850276350753917),doc(ar,0.18039
  756921425887),doc(ln,0.179824892
  36143614),doc(mkstr,0.1691029950
  2983564),doc(indxbib,0.169015292
  83334107),doc(ctags,0.1558855062
  5426306),doc(clear_colormap,0.15
  131210125256 64),doc(du,0.1487652
  41020429 96),doc(whois,0.14317701
  087473347),doc(iconedit,0.131044
  54711216584)]}).
query_doc_rel_th(reference(10),
  docRel([doc(ed,0.505202627595207
  7),doc(red,0.505202627595207 7),d
  oc(swin,0.435131979029601 1),doc(
  dbx,0.3338501454352308),doc('scc
  s-
  unget',0.3141533121049456),doc(u
  nget,0.3141533121049456),doc(scr
  eenblank,0.2860257245013854),doc
  (indent,0.282022394481179 66),doc
  (logname,0.263439928411391 24),do
  c(gcore,0.2604819611689 41),doc(m
  kstr,0.2575302244625154),doc(whe
  reis,0.2478087354783928),doc(dbx
  tool,0.2337326048447350 4),doc(sc
  cs,0.2260087636030394 5),doc(erro
  r,0.2160045416070987 5),doc(su,0.
  1824063856267539 3),doc(finger,0.
  1589828553693257),doc(talk,0.13
  799005296119746),doc(write,0.137
  977620720840 55),doc(last,0.12827
  500735237224),doc(groups,0.12586
  681609314884),doc(wall,0.1247396
  56846405 66),doc(users,0.12368288
  391604497),doc(rwall,0.108072186
  24168337),doc(id,0.1066698949423
  3982),doc(syswait,0.105943412422
  80068),doc(quota,0.1055763522463
  93174),doc(crontab,0.100957902503
  92196),doc(telnet,0.096655505372
  59007),doc(whois,0.089402498106 1
  052)]}).
query_doc_rel_th(reference(11),
  docRel([doc(finger,0.624135556222
  6522),doc(graph,0.401081041526 1
  3444),doc(vfontinfo,0.393416408 1
  289623),doc(mkdir,0.376888988034
  367),doc(delta,0.363542688716250
  73),doc('sccs-
  comb',0.363542688716250 73),doc(l
  ook,0.3306355192859343),doc(enro
  ll,0.3029265312421504 7),doc(xget
  ,0.3029265312421504 7),doc(xsend,
  0.3029265312421504 7),doc(env,0.2
  9688724639347 25),doc('sccs-
  unget',0.2742377062470736),doc(u
  nget,0.2742377062470736),doc(fin
  d,0.2621764024576645),doc(lpstat
  ,0.2610347811645560 6),doc(whatis
  ,0.2570927107010741 5),doc(what,0
  .2540023107201272 4),doc(overview
  ,0.2537624523628776 4),doc(prt,0.
  2530052606404491 5),doc('sccs-
  prt',0.2530052606404491 5),doc('o
  id-
  syslog',0.2519981393939972 3),doc
  (sorthib,0.2505825911401794 6),do
  c(lookbib,0.2478967040384716 7),d
  oc(cluster,0.2450253068360804 5),
  doc(swin,0.2440536171380575 4),do
  c(leave,0.2424772155990903 3),doc
  (from,0.2423416231033480 6),doc(t
  sort,0.2404075270331409 2),doc(so
  rt,0.2245735786255459 4),doc(stri
  ngs,0.2242274391231445 5),doc(upt
  ime,0.2220941937054659),doc(ln,0
  .2152248924962949 5),doc(logname,
  0.2067060154078092),doc(script,0
  .2059371638550861),doc(gcore,0.2
  0438507042145274),doc(vwidth,0.1
  9691564131975822),doc(chfn,0.192
  58230014224 98),doc(chsh,0.192582
  30014224 98),doc(passwd,0.1925823
  0014224 98),doc(lorder,0.19055249
  151443665),doc(mach,0.1870037388
  1297985),doc(comm,0.1736691989 42
  94178),doc(file,0.16929821274533
  172),doc(clear_colormap,0.160849
  78974638992),doc(i,0.135053442861
  26533),doc(iapx,0.135053442861 26
  533),doc(m,0.135053442861 26533)
  ,doc(machid,0.135053442861 26533)
  ,doc(pdp,0.135053442861 26533),do
  c(sparc,0.135053442861 26533),doc
  (sun,0.135053442861 26533),doc(u3
  b15,0.135053442861 26533),doc(u3b
  2,0.135053442861 26533),doc(u3b5,
  0.135053442861 26533),doc(vax,0.1
  35053442861 26533),doc(cp,0.06836
  041720676837),doc(rm,0.049468590
  471679244),doc(rmdir,0.049468590
  471679244),doc(mkstr,0.049217547
  35681729),doc(rcp,0.048608403918
  39152),doc(install,0.046558039777
  131251),doc(pack,0.045858974643 7
  0337),doc(pcat,0.045858974643703
  37),doc(unpack,0.045858974643703
  37),doc(rmdel,0.043171800815161 9
  7),doc('sccs-
  rmdel',0.043171800815161 97),doc(
  'sccs-
  val',0.041621061896692485),doc(v
  al,0.041621061896692485),doc(ftp
  ,0.041249621975283 36),doc(usenet
  ,0.037407809868525 45),doc(size,0
  .037011721649057694),doc(crypt,0
  .036164520074099 47),doc(organize
  r,0.035605248993046),doc(get,0.
  034745156675101 55),doc('sccs-
  get',0.034745156675101 55),doc('uu
  pick,0.034061971558777 91),doc(uu
  to,0.034061971558771 91),doc(tail
  ,0.033559860909648444),doc(rdist
  ,0.033005591778296706),doc(cpio,0
  .032938114665230 13),doc(paxcpio,
  0.032938114665230 13),doc(split,0
  .031500345388908196),doc(chgrp,0
  .031311838465023 28),doc(compress
  ,0.031191298324583457),doc(uncom
  press,0.031191298324583457),doc(
  zcat,0.031191298324583457),doc('
  sccs-
  sccsdiff',0.031065388129298 91),d
  oc(sccsdiff,0.031065388129298 91)
  ,doc(admin,0.030145942496386125)
  ,doc('sccs-
  admin',0.030145942496386125),doc
  (mv,0.029800867551836755),doc(su
  m,0.029800867551836755),doc(tftp
  ,0.029693527173703077),doc(chmod
  ,0.029687424102748133),doc(dd,0.
  029442507935950348),doc(vswap,0.
  029423963470795 78),doc(egrep,0.
  028078558480107 19),doc(fgrep,0.0
  280785584801071 9),doc(grep,0.028
  07855848010719),doc(more,0.02777
  632664228251),doc(page,0.0277763
  2664228251),doc('old-
  ccat',0.027626685436731 39),doc('
  old-
```

```
compact',0.027626685436731139),doc('old-
uncompact',0.027626685436731139),doc(acctcom,0.027498651566885521),doc(sact,0.027390259637910941),doc('sccs-
show',0.027390259637910941),doc(indent,0.026949167976360825),doc(bar,0.026926212602078035),doc(tar,0.026926212602078035),doc(ar,0.026252420615882635),doc(screendump,0.026014346257232427),doc(screenload,0.026014346257232427),doc(diff,0.025711235480329813),doc(touch,0.025412507155884077),doc(cut,0.025328859864351835),doc(head,0.025158823276080451),doc('old-
filemerge',0.025114785615961255),doc(checknr,0.024966742402079088),doc(dos2unix,0.024412492243244447),doc(unix,0.024412492243244447),doc(pg,0.023991282684827888),doc(make,0.023767354983424294),doc(objdump,0.023713155893805421),doc(diffmk,0.023700289764575395),doc(strip,0.023503757920488873),doc(df,0.023231144883674285),doc(uudecode,0.023044884835412161),doc(uuencode,0.023044884835412161),doc(ctags,0.022685291909066987),doc(csplit,0.022459232846628666),doc(crontab,0.022344282326000971),doc(paste,0.021118110521745811),doc(error,0.020640710770984511),doc(du,0.020559091718686237),doc(cmp,0.020536423220577051),doc(sdiff,0.019893900821437281),doc(pr,0.019712425972024)}]}).
query_doc_rel_th(reference(12),
  docRel([doc(uudecode,0.467748889910385 6),doc(uuencode,0.467748889910385 6),doc(tty,0.464132559952517 87),doc(id,0.46070353312086031),doc(uname,0.40770017083650417),doc(diffmk,0.38547106560549821),doc(sdiff,0.32356242159332561),doc(chkey,0.31818514485625354),doc(nm,0.30771207921060721),doc(find,0.29426974943085511),doc(hostname,0.28243422545000971),doc(keylogin,0.27766518199330771),doc(disablenumlock,0.27675583285061681),doc(enablenumlock,0.27675583285061681),doc(keylogout,0.26664733404847605),doc(domainname,0.25487293060814087),doc(strings,0.23629568657180721),doc(logname,0.22696952505131961),doc(ypmatch,0.22033064959270131),doc(whereis,0.21558487157476151),doc(cd,0.21406750107107141),doc(whois,0.19306312605622141),doc(clear_functions,0.19150927517663271),doc(cp,0.06348303519896471),doc(rm,0.045939103336043584),doc(rmdir,0.045939103336043584),doc(mkstr,0.045705971656214381),doc(rcp,0.045140289410217596),doc(install,0.043236214733311021),doc(pack,0.042587026534507556),doc(pcat,0.042587026534507556),doc(unpack,0.042587026534507556),doc(rmdel,0.040091577300676081),doc('sccs-
rmdel',0.040091577300676081),doc('sccs-
val',0.038651480569729631),doc(val,0.038651480569729631),doc(ftp,0.038306542159921591),doc(delta,0.037748771174543011),doc('sccs-
comb',0.037748771174543011),doc(usend,0.034738835829759321),doc(size,0.034371007729710161),doc(crypt,0.033584252329437961),doc(organizer,0.033065233231781451),doc(get,0.032266157731708911),doc('sccs-
get',0.032266157731708911),doc(uupick,0.031631718393954261),doc(uuto,0.031631718393954261),doc(tail,0.031165430499868761),doc(rdist,0.030697442729217428),doc(what,0.030694279384354691),doc(cpio,0.030588044633427267),doc(paxcpio,0.030588044633427267),doc(prt,0.030573793339881045),doc('sccs-
prt',0.030573793339881045),doc(split,0.029252857381707311),doc(chgrp,0.029077800057992641),doc(compress,0.028965860220713941),doc(uncompress,0.028965860220713941),doc(zcat,0.028965860220713941),doc('sccs-
sccsdiff',0.028848933471496183),doc(sccsdiff,0.028848933471496183),doc(admin,0.027995088485424948),doc('sccs-
admin',0.027995088485424948),doc(mv,0.027674633962965725),doc(sum,0.027674633962965725),doc(tftp,0.027574952110780421),doc(chmod,0.027569284481996317),doc(dd,0.027341842604754161),doc(vswap,0.027324621135734636),doc(egrep,0.026075208273485306),doc(fgrep,0.026075208273485306),doc(grep,0.026075208273485306),doc(more,0.025794540085916491),doc(page,0.025794540085916491),doc('sccs-
unget',0.025756059744737761),doc(unget,0.025756059744737761),doc('old-
ccat',0.025655575487580528),doc('old-
compact',0.025655575487580528),doc('old-
uncompact',0.025655575487580528),doc(acctcom,0.025536676583825742),doc(sact,0.025436018208566461),doc('sccs-
show',0.025436018208566461),doc(indent,0.025026397572318201),doc(bar,0.025005080019074012),doc(tar,0.025005080019074012),doc(ar,0.024379361772694041),doc(file,0.024312001988169081),doc(screendump,0.024158273553696661),doc(screenload,0.024158273553696661),doc(cluster,0.023900570305610665),doc(diff,0.023876789137633301),doc(touch,0.023599374494619123),doc(cut,0.023521695273666191),doc(head,0.023363790463259683),doc('old-
filemerge',0.023322894804022103),doc(checknr,0.023185414183776273),doc(dos2unix,0.022670708688928076),doc(unix,0.022670708688928076),doc(ln,0.022348063845260013),doc(pg,0.022279551608682183),doc(make,0.022071600710617447),doc(objdump,0.022021268611999226),doc(strip,0.021826810782345551),doc(df,0.021573930418901451),doc(ctags,0.021066742576661341),doc(csplit,0.020856810528697381),doc(crontab,0.020750061502802456),doc(paste,0.019611374657539091),doc(error,0.019168036444872995),doc(du,0.019092240752253326),doc(cmp,0.019071393074772795),doc(pr,0.018305982949956081)]).
query_doc_rel_th(reference(13),
  docRel([doc(chfn,0.65297578581409591),doc(chsh,0.65297578581409591),doc(passwd,0.65297578581409591),doc(yppasswd,0.59922927228460171),doc(wc,0.43804561608374826),doc(apropos,0.42687645300929294),doc(mv,0.42290610903080184),doc(pax,0.38943394441630711),doc(whatis,0.38249418220499604),doc(stty,0.36401652394044026),doc(touch,0.36063059249259851),doc(look,0.35100424101249744),doc(cd,0.29049557845703242),doc(env,0.28453915839036561),doc(dd,0.28321678447611161),doc(vswap,0.28303839821768867),doc(coloredit,0.27181138103156671),doc(chgrp,0.26154745004964763),doc(ranlib,0.25724611639829891),doc(chkey,0.25567462940128366),doc(chmod,0.24797873434642634),doc(cdc,0.24358752845516981),doc('sccs-
cdc',0.24358752845516981),doc(dos2unix,0.23483147458839773),doc(unix,0.23483147458839773),doc(sv_acquire,0.19792618993800431),doc(sv_release,0.19792618993800431),doc('old-
sun3cvt',0.18662981606848197),doc(rasfilter,0.16761857444994241)]).
query_doc_rel_th(reference(14),
  docRel([doc(tail,0.48632535045473621),doc(wait,0.40962455598929337),doc(find,0.37232844061600521),doc(lastcomm,0.35580696480876331),doc(last,0.32736710030984391),doc(sccs,0.30439525182887441),doc(tcolplaces,0.29952026601584551),doc(sact,0.13527316801392341),doc('sccs-
show',0.13527316801392341),doc(rup,0.11243564403575522),doc(ruptime,0.11243564403575522),doc(size,0.10956514779791229),doc(mpstat,0.10386409547436203),doc(prt,0.09746069181060604),doc('sccs-
prt',0.09746069181060604),doc(compress,0.09233505128451548),doc(uncompress,0.09233505128451548),doc(zcat,0.09233505128451548),doc(tty,0.09119079509264143),doc(uname,0.08010319883998218),doc(ps,0.07893837465587829),doc(cal,0.07779340047852808),doc(cat,0.07779340047852808),doc(diff,0.07611251772720849),doc(lpstat,0.07485861456428241),doc(head,0.07447722160459229),doc('old-
prmail',0.07360351479794632),doc(date,0.07193276620208239),doc(uptime,0.07141992430144131),doc(cp,0.07131606885340483),doc(prof,0.06850564131343838),doc(nroff,0.06849942226373383),doc(finger,0.06754519110368109),doc(mach,0.06557757270998342),doc(lpq,0.06424029625404942),doc(du,0.06086071725689547),doc(arch,0.05946660258930554),doc(clock,0.05923754066031996),doc(sdiff,0.05889156435494394),doc(man,0.05823152378304667),doc(pagesize,0.05715158748020496),doc(basename,0.05683128527627347),doc(dirname,0.05683128527627347),doc(prs,0.05653138205844571),doc('sccs-
prs',0.05653138205844571),doc(fold,0.05632998812776481),doc(pwd,0.05525765399125959),doc(groups,0.05340352106920892),doc(users,0.05247690934909272),doc(whoami,0.05247507114473963 5),doc(rm,0.05167424939131543 5),doc(rmdir,0.05167424939131543 5),doc(vedit,0.05154704644131632 85),doc(vi,0.05154704644131632 85),doc(view,0.05154706441316328 5),doc(traffic,0.05142022410760202 5),doc(mkstr,0.05134553210051166),doc(wc,0.05106609177366576),doc(mps,0.05074862161533 1925),doc(rcp,0.05071005515523901 53),doc(which,0.05050143649305469),doc(domainname,0.05007635144607324 6),doc(banner,0.04930402788 68403114),doc(gprof,0.04925875246525941),doc(install,0.04857103724173491 4),doc(printenv,0.04821145723399377),doc(pack,0.04784174712311565),doc(pcat,0.04784174712311565),doc(unpack,0.04784174712311565),doc(comm,0.04776899644483),doc(col,0.04506415844159017 4),doc(rmdel,0.04503838983526176),doc('sccs-
rmdel',0.04503838983526176),doc(quota,0.04479456243199119 6),doc(whatis,0.04459006618077697 4),doc('sccs-
val',0.04342060270051217),doc(val,0.04342060270051217),doc(ftp,0.04303310309047645 6),doc(delta,0.04240650995620695),doc('sccs-
comb',0.04240650995620695),doc(atq,0.04219488606055691 5),doc(uusend,0.03902518523504126),doc(from,0.03835128789319288),doc(crypt,0.03772814018752742 4),doc(organizer,0.03714508045214066),doc(perfmeter,0.03650659847720476),doc(get,0.03624740876389128 5),doc('sccs-
get',0.03624740876389128 5),doc(upick,0.03553468733598835),doc(uuto,0.03553468733598835),doc(rdist,0.03448513342878135),doc(what,0.03448157976567543),doc(cpio,0.03436223693986348),doc(paxcpio,0.03436223693986348),doc(split,0.03286230383683185),doc(chgrp,0.03266564657064726 5),doc('sccs-
sccsdiff',0.03240854063376713),doc(sccsdiff,0.03240854063376713),doc(admin,0.03144934157175089 5),doc('sccs-
```

```
admin',0.031449341571750895),doc
(mv,0.031089346862668996),doc(su
m,0.031089346862668996),doc(tftp
,0.0309773654835243),doc(chmod,
0.030970998538349948),doc(dd,0.0
30715493102187488),doc(vswap,0.0
30691467113870904),doc(egrep,0.0
29292571515690157),doc(fgrep,0.0
29292571515690157),doc(grep,0.02
9292571515690157),doc(more,0.028
97727229083607),doc(page,0.02897
727229083607),doc('sccs-
unget',0.028939211527743494),doc
(unget,0.028939211527743494),doc
('old-
ccat',0.02882116115292257),doc('
old-
compact',0.02882116115292257),do
c('old-
uncompact',0.02882116115292257),
doc(acctcom,0.02868759157200705)
,doc(indent,0.02811435034299107)
,doc(bar,0.02809040246321211),do
c(tar,0.02809040246321211),doc(a
r,0.02738747820318225),doc(file,0
.027311807041334665),doc(screend
ump,0.027139110389651266),doc(scr
eenload,0.027139110389651266),doc
(cluster,0.02684960969822013),do
c(touch,0.026511249991111327),do
c(cut,0.026423986100036708),doc(
'old-
filemerge',0.026200656072781523)
,doc(checknr,0.02604621201778743
7),doc(dos2unix,0.025467998131674
028),doc(unix,0.025467998131674
028),doc(ln,0.0251055428420165),
doc(pg,0.025028577029563227),doc(
make,0.024794967522424436),doc(ob
jdump,0.024738425064595888),doc(d
iffmk,0.024725002651516224),doc(s
trings,0.024570601713868878),doc
(strip,0.024519973497256146),doc
(df,0.024235890775714385),doc(uu
decode,0.024041260436870045),doc(
uuencode,0.024041260436870045),do
c(ctags,0.023666122128619227),doc
(csplit,0.023430287021803553),do
c(crontab,0.023310366465754264),
doc(paste,0.022031179526995657),
doc(error,0.021533138776411696),
doc(cmp,0.021424570791048452),doc
(pr,0.020564718375494638)]}}.
query_doc_rel_th(reference(15),
  docRel([doc(time,0.8404225029980
488),doc(clock,0.6364023477832059
),doc(date,0.5283310126610505),
doc(at,0.4415073350246928),doc(b
atch,0.4415073350246928),doc(tou
ch,0.43449937216886675),doc(atq,
0.37633005410064113),doc(hostname
,0.33443767876009317),doc(domain
name,0.30180163595808899),doc(tty
,0.27479569045694646),doc(id,0.272
7654907404016),doc(uname,0.24138
41639542861),doc(ul,0.23706315066
817132),doc(from,0.223732524996
98174),doc(file,0.21042839493255
738),doc(perfmeter,0.19114946799
97987),doc(nm,0.18218491992899843
),doc(stty,0.18112627388922903)
,doc(pr,0.17898150493597512),doc
(dos2unix,0.17078608278149585),d
oc(unix,0.17078608278149585),doc
(find,0.15908506660596639),doc(s
win,0.1586600415164481),doc(stty
_from_defaults,0.15211817867802163
),doc(tabs,0.15045501330480643
),doc(printenv,0.1452811181822211
),doc(foption,0.14098832722513727
),doc(logname,0.13438024550052
2),doc(whois,0.11430552304617606
),doc(ipcrm,0.11197663936883011)
]}}).
query_doc_rel_th(reference(16),
  docRel([doc(rcp,0.792241882191422
27),doc(uusend,0.710794571769284
7),doc(cp,0.5633320419058909),do
c(rsh,0.382315793406227746),doc(u
ucp,0.348063938740612),doc(uulog
,0.348063938740612),doc(uuname,0
.348063938740612),doc(rlogin,0.3
360358912533029),doc(mach,0.3307
5639361877346),doc(rdist,0.30848
270629720037),doc(cu,0.307910728
7973861),doc(tip,0.307910728797
3861),doc(arch,0.2999342316165051
),doc(uupick,0.280691691189790
15),doc(uuto,0.28069169118979015),
doc(uux,0.27996405765503491),doc(
hostname,0.27988472686144733),do
c(rup,0.27907869399901132),doc(ru
ptime,0.27907869399901132),doc(cp
io,0.27143040006282054),doc(paxc
pio,0.27143040006282054),doc(dd,
0.24262444250774895),doc(hostid,
0.23218982173813912),doc(tcopy,0
.23180084112069831),doc(on,0.2099
3249812168263),doc(pg,0.19770298
0979575),doc(get,0.17721369744885
923),doc(telnet,0.17237377649688
6889),doc(rm,0.06303832382174694
),doc(rmdir,0.06303832382174694)
,doc(mkstr,0.062718417048236841),
doc(install,0.059329379705896891)
,doc(pack,0.058438553962399211),d
oc(pcat,0.058438553962399211),doc
(unpack,0.058438553962399211),doc
(rmdel,0.055014261247491661),doc(
'sccs-
rmdel',0.055014261247491661),doc(
'sccs-
val',0.053038139001570181),doc(va
l,0.053038139001570181),doc(ftp,0
.052564809363048984),doc(delta,0
.051799427685103115),doc('sccs-
comb',0.051799427685103115),doc(
size,0.047164410230151846),doc(c
rypt,0.046084812717562804),doc(o
rganizer,0.045372606958813705),d
oc(get,0.04427610362127291),doc('
sccs-
get',0.04427610362127291),doc(tai
l,0.042765669271421051),doc(what,
0.042119148672803911),doc(prt,0.0
41953815922794115),doc('sccs-
prt',0.041953815922794115),doc(sp
lit,0.040141207869257914),doc(ch
grp,0.039900991594704831),doc(com
press,0.039747386077868554),doc(
uncompress,0.039747386077868554)
,doc(zcat,0.039747386077868554),
doc('sccs-
sccsdiff',0.039586937445976561),d
oc(sccsdiff,0.039586937445976561)
,doc(admin,0.038415278601618951),
doc('sccs-
admin',0.038415278601618951),doc(
mv,0.037975546119050484),doc(sum
,0.037975546119050484),doc(tftp,
0.037838761192465331),doc(chmod,0
.037830983987586511),doc(vswap,0.0
37495252328914131),doc(egrep,0.035
780790296222297),doc(fgrep,0.035
780790296222297),doc(grep,0.0357
80790296222297),doc(more,0.03539
56532167063261),doc(page,0.035395
653216706326),doc('sccs-
unget',0.035349162106084644),doc
(unget,0.035349162106084644),doc
('old-
ccat',0.035204963919059806),doc('
old-
compact',0.035204963919059806),d
oc('old-
uncompact',0.035204963919059806)
,doc(find,0.035092399500445577),d
oc(acctcom,0.035041809067252567),
doc(sact,0.034903684141111824),do
c('sccs-
show',0.034903684141111824),doc(i
ndent,0.034341595585272961),doc(b
ar,0.034312344320276314),doc(tar
,0.034312344320276314),doc(ar,0.
033453724395809285),doc(file,0.0
33361292293284655),doc(screendump
,0.033150343839154085),doc(scree
nload,0.033150343839154085),doc(
cluster,0.032796719592639577),doc
(diff,0.032764087617870534),doc(t
ouch,0.032383414201627686),doc(c
ut,0.032276821614288805),doc(head
,0.032060142274715331),doc('old-
filemerge',0.032004024640222889),
doc(checknr,0.031815371679478841)
,doc(dos2unix,0.031109085111421624),doc(unix,0.031109085111421624),doc(ln,0.030666347036888083),
doc(make,0.030286980193811295),d
oc(objdump,0.030217913747210311),
doc(diffmk,0.030201518308953394)
,doc(strings,0.030012918015922062),doc(strip,0.029951075797640836),doc(df,0.029604069585474681),d
oc(uudecode,0.029366329196730472),doc(uuencode,0.029366329196730472),doc(ctags,0.028908098852138662),doc(csplit,0.028620026960022922),doc(crontab,0.028473544352106181),doc(paste,0.026911021253686093),doc(error,0.026302665593581744),doc(du,0.026198657954192261),doc(cmp,0.026170050459737356),
doc(sdiff,0.025350998484127821),d
oc(pr,0.025119743253005641)]}}.
query_doc_rel_th(reference(17),
  docRel([doc(indent,0.561600414669
4601),doc(sccs,0.413733604475398
886),doc(crypt,0.41080523856211104),doc(mkstr,0.34376550587721943
),doc(rpcgen,0.34101890263571644
),doc(dos2unix,0.33542878366997275
),doc(unix,0.33542878366997275
),doc(dis,0.28303022719159641),do
c(cc,0.26982192876917921),doc(des
,0.26967854057828481),doc(telnet,
0.26293812104837471),doc(uudecode
,0.26177478349084904),doc(uuenco
de,0.26177478349084904),doc(troff,0.26169622912215281),doc(lex,0.
26041485770751451),doc(dbx,0.2487
6698670541403),doc(fdformat,0.24
79547546407228),doc(nroff,0.2454
77994808431381),doc(roffbib,0.227
73477451135811),doc(tbl,0.2133714
25615571),doc(lint,0.209041097827
96168),doc(yacc,0.20616454670577
614),doc(dd,0.20227071711373991),
doc(foption,0.19925086619288383)
,doc(cpp,0.19365730023496771),doc
(whereis,0.18465360356177732),do
c(cb,0.17845110993139032),doc(db
xtool,0.17416483592130866),doc(s
trace,0.16416714848963262),doc(a
rror,0.16095484655317134),doc(cxr
ef,0.15987601782736452),doc(cflo
w,0.15770294321966116),doc(xstr,
0.13479735621699984),doc(csh,0.1
0967871779184045)]}}.
query_doc_rel_th(reference(18),
  docRel([doc(kill,0.87362093705788
21),doc(wait,0.7989310532671012
),doc(sync,0.5242265439119137),do
c(tabs,0.5177236438626871),doc(ch
ecknr,0.49630046973636779),doc(no
hup,0.44076422382125714),doc(scc
s,0.42930885965418397),doc(cd,0.
3786954096563702),doc(du,0.36178
486540605072),doc(pwd,0.31387159
4538312),doc(ps,0.31342219387875
375),doc(inline,0.29935911267382
26),doc(ustar,0.2801363033772298
3),doc(acctcom,0.2093078512535318),doc(mps,0.20149571602327923
),doc(gcore,0.17507096873692543)]
}}).
query_doc_rel_th(reference(19),
  docRel([doc(tail,0.3533294479877
57734),doc(split,0.33164617927503
87),doc(test,0.2598357626445401)
,doc(objdump,0.24966004193411523
),doc(csplit,0.23645832041116593
),doc(awk,0.22072785032030273),d
oc(nawk,0.22072785032030273),doc
(lptest,0.19938062643939962),doc(
cp,0.05162495940417251),doc(rm,0.
037358080585693342),doc(rmdir,0.0
3735808058569342),doc(mkstr,0.03
7168495864848418),doc(rcp,0.036708
478115316781),doc(install,0.03516
006793612547),doc(pack,0.03463214
24215651611),doc(pcat,0.034632142
41565161),doc(unpack,0.034632142
41565161),doc(rmdel,0.0326028212
751617),doc('sccs-
rmdel',0.0326028212751617),doc('
sccs-
val',0.0314317220194285),doc(val
,0.0314317220194285),doc(ftp,0.0
31151215087970856),doc(delta,0.0
30697630844767187),doc('sccs-
comb',0.030697630844767187),doc(
uusend,0.028249925099497841),doc(
size,0.027950804071774371),doc(cr
ypt,0.027311007699817119),doc(org
anizer,0.026888936830607861),doc(
get,0.026239121645766292),doc('s
ccs-
get',0.026239121645766292),doc(u
upick,0.025723190027920027),doc(
uuto,0.025723190027920027),doc(r
dist,0.024963428886802881),doc(wh
at,0.024960856427095991),doc(cpio
,0.024874465398583383),doc(paxcp
io,0.024874465398583383),doc(prt
,0.024862876122039274),doc('sccs
-
```

```
prt',0.024862876122039274),doc(chgrp,0.023646321302246044),doc(compress,0.023555290847619694),doc(uncompress,0.023555290847619694),doc(zcat,0.023555290847619694),doc('sccs-sccsdiff',0.023460204992592284),doc(sccsdiff,0.023460204992592284),doc(admin,0.022765850782759188),doc('sccs-admin',0.022765850782759188),doc(mv,0.022505254362613372),doc(sum,0.022505254362613372),doc(tftp,0.02242419220865515),doc(chmod,0.022419583243372603),doc(dd,0.022234625519517322),doc(vswap,0.022206208705298154),doc(egrep,0.021204587404415324),doc(fgrep,0.021204587404415324),doc(grep,0.021204587404415324),doc(more,0.020976345579747194),doc(page,0.020976345579747194),doc('sccs-unget',0.020948793789791064),doc(unget,0.020948793789791064),doc('old-ccat',0.020863338353088498),doc('old-compact',0.020863338353088498),doc('old-uncompact',0.020863338353088498),doc(find,0.020796629875344773),doc(acctcom,0.020766648724744434),doc(sact,0.020684792453692334),doc('sccs-show',0.020684792453692334),doc(indent,0.020351685370025884),doc(bar,0.020334349749298576),doc(tar,0.020334349749298576),doc(ar,0.019825510199227046),doc(file,0.019770732633367507),doc(screendump,0.019645719326047964),doc(screenload,0.019645719326047964),doc(cluster,0.019436152991003344),doc(diff,0.019416813737229883),doc(touch,0.019191217723446917),doc(cut,0.019128048301637224),doc(head,0.018999638728804284),doc('old-filemerge',0.018966382021146058),doc(checknr,0.018854581578148146),doc(dos2unix,0.018436018569052706),doc(unix,0.018436018569052706),doc(ln,0.018173640960257965),doc(pg,0.018117926210310454),doc(make,0.017948818721404053),doc(diffmk,0.017898171880118194),doc(strings,0.017786402649618456),doc(strip,0.017749753410963555),doc(df,0.017544108887887464),doc(uudecode,0.017403218012896974),doc(uuencode,0.017403218012896974),doc(xget,0.017116659299050497),doc(crontab,0.016874131480306552),doc(paste,0.015948141179541092),doc(error,0.015587614867860084),doc(du,0.015525977147763085),doc(cmp,0.015509023634116003),doc('sdiff,0.015023633035926642),doc(pr,0.014886585426847523)]}).
query_doc_rel_th(reference(20),
  docRel([doc(mail,0.7081290741508339),doc(enroll,0.691166068363192 7),doc(xget,0.691166068363192 7),doc(xsend,0.691166068363192 7),doc(uusend,0.47352308430197 14),doc(write,0.46122337218780 28),doc('old-prmail',0.41720871324498315),doc(wall,0.4169722950407 77),doc(mailtool,0.38629326745995 05),doc(rwall,0.36125726706576 44),doc(cancel,0.33219604771267 5),doc(lp,0.33219604771267 5),doc(vacation,0.30346019209820 36),doc(fmt,0.29891028544317544),doc(fmt_mail,0.29891028544317544),doc(kill,0.2923312022289976),doc(bin,0.2895077701499429),doc(biff,0.24239977 32741047),doc(strings,0.23223622 69168536),doc(uudecode,0.22723300 30498502),doc(uuencode,0.22723300 30498502),doc(from,0.21738760053960554),doc(whereis,0.21188121493560338),doc('old-filemerge',0.19231976572586043),doc(textedit,0.18963381039023338),doc(vedit,0.1813631406555254),doc(vi,0.1813631406555254),doc(view,0.1813631406555254),doc(ypcat,0.17964703244818875),doc(cp,0.06239242358474938),doc(rm,0.045149889028821166),doc(rmdir,0.045149889028821166),doc(mkstr,0.044920766245233533),doc(rcp,0.044364798387371424),doc(install,0.04249343490548105),doc(pack,0.041855399484773178),doc(pcat,0.041855399484773178),doc(unpack,0.041855399484773178),doc(rmdel,0.039402821010128 67),doc('sccs-rmdel',0.039402821010128 67),doc('sccs-val',0.037987464530108 17),doc(val,0.037987464530108 17),doc(ftp,0.037648452016852 34),doc(delta,0.037160026329938 7336),doc('sccs-comb',0.037100263299387336),doc(size,0.033780528397655 94),doc(crypt,0.033007289121386174),doc(organizer,0.032497186551651985),doc(get,0.031711838829690184),doc('sccs-get',0.031711838829690184),doc(upick,0.031088298890619016),doc(uuto,0.031088298890619016),doc(tail,0.030630021624747 24),doc(rdist,0.030170073685467 91),doc(what,0.030166964685527976),doc(cpio,0.030062555002555037),doc(paxcpio,0.030062555002555037),doc(prt,0.030048548540186317),doc('sccs-prt',0.030048548540186317),doc(split,0.028750305701412165),doc(chgrp,0.028578255788256585),doc(compress,0.028468239029895 96),doc(uncompress,0.028468239029895 96),doc(zcat,0.028468239029895 96),doc('sccs-sccsdiff',0.028353321032627 59),doc(sccsdiff,0.028353321032627 59),doc(admin,0.027514144741202 58),doc('sccs-admin',0.027514144741202 58),doc(mv,0.027199195491497352),doc(sum,0.027199195491497352),doc(tftp,0.027101226131245982),doc(chmod,0.027095655869920 09),doc(dd,0.026872121347370576),doc(vswap,0.026855195736157137),doc(egrep,0.025627247256860 65),doc(fgrep,0.025627247256860 65),doc(grep,0.025627247256860 65),doc(more,0.025351400829690403),doc(page,0.025351400829690403),doc('sccs-unget',0.025131810253814 11),doc(unget,0.025131810253814 11),doc('old-ccat',0.025214823584202885),doc('old-compact',0.025214823584202885),doc('old-uncompact',0.025214823584202885),doc(find,0.025134201659302267),doc(acctcom,0.025097967313175817),doc(sact,0.024999038206102508),doc('sccs-show',0.024999038206102508),doc(indent,0.024596454678617583),doc(bar,0.024575503351894 89),doc(tar,0.024575503351894 89),doc(ar,0.023960534679548 19),doc(file,0.023894321075462 95),doc(screendump,0.023743244662364087),doc(screenload,0.023743244662364087),doc(cluster,0.023489968646758 44),doc(diff,0.023466596029158585),doc(touch,0.023193947252027728),doc(cut,0.023117602527138015),doc(head,0.022962410454388 59),doc(checknr,0.022787098603716676),doc(dos2unix,0.022281235531270427),doc(unix,0.022281235531270427),doc(ln,0.021964133588778 93),doc(pg,0.021896798363381 54),doc(make,0.021692419973529508),doc(objdump,0.021642952566869596),doc(diffmk,0.021631209675038086),doc(strip,0.021451835412381 58),doc(df,0.021203299446305 62),doc(ctags,0.020704824876038128),doc(csplit,0.020498499171601 71),doc(crontab,0.020393584248687043),doc(paste,0.019274459560376 42),doc(error,0.018838737710132 88),doc(du,0.018764244155358687),doc(cmp,0.018743754632133 36),doc(sdiff,0.018157125680637508),doc(pr,0.017991493928561 97)]}}.

query_doc_rel_th(reference(21),
  docRel([doc(egrep,0.85313158560115 51),doc(fgrep,0.85313158560115 51),doc(grep,0.85313158560115 51),doc(acctcom,0.30409097789326284),doc(which,0.28221214999844246),doc(apropos,0.27809133573405 62),doc(whereis,0.25367806014985 28),doc(test,0.17944084995160292),doc(expr,0.16849264845222522),doc(cpio,0.13516346277354457),doc(paxcpio,0.13516346277354457),doc(login,0.13472341505146 23),doc(look,0.13380413690697224),doc(gfxtool,0.10214861766974064),doc(shelltool,0.09701792959118449),doc(clock,0.09685251846337514),doc(strings,0.09664817851900014),doc(newgrp,0.09404591973354355),doc(who,0.09226083082652987),doc(lookbib,0.09119199313974867),doc(wall,0.08653206928047 21),doc(users,0.08579898447805313),doc(getopt,0.08454748007718697),doc(getopt,0.08454748007718697),doc(getopts,0.08454748007718697),doc(ypcat,0.08348123127494 75),doc(ypasswd,0.08286471075097948),doc(pr,0.08089108259916565),doc(banner,0.08061136934810 27),doc(rev,0.07890139911091 02),doc(comm,0.07809998011965384),doc(inline,0.07627996281581 98),doc(rusers,0.07617311113732851),doc(rwho,0.07617311137328 51),doc(logname,0.07291055655045052),doc(lastcomm,0.07249574829776943),doc(cp,0.06625599726533 7864),doc(perfmeter,0.05968775819581232),doc(w,0.05789448643398783),doc(xm,0.04779457226752 4451),doc(rmdir,0.04794572226752 451),doc(mkstr,0.04770240740151400 5),doc(rcp,0.04711201616858475),doc(install,0.04512477155526955 4),doc(pack,0.04444722683172527),doc(pcat,0.04444722683172527),doc(unpack,0.04444722683172527),doc(rmdel,0.04184277643523446),doc('sccs-rmdel',0.04184277643523446),doc('sccs-val',0.04033977631363315),doc(val,0.04033977631363315),doc(ftp,0.03997977100236096),doc(delta,0.03939736593723896),doc('sccs-comb',0.03939736593723896),doc(uusend,0.03625622734529878 5),doc(size,0.03587233252268580 4),doc(crypt,0.03505121166538525),doc(organizer,0.03450952182599683),doc(get,0.03367554272725359),doc('sccs-get',0.03367554272725359),doc(uupick,0.03301339109444821),doc(uuto,0.03301339109444821),doc(tail,0.03252673575633691),doc(rdist,0.03203830629109217),doc(what,0.03203500477140172),doc(prt,0.03190925589939635 3),doc('sccs-prt',0.03190925589939635 3),doc(split,0.03053062149026338 3),doc(chgrp,0.03034791766849092),doc(compress,0.03023108830016388 7),doc(uncompress,0.03023108830016388 7),doc(zcat,0.03023108830016388 7),doc('sccs-sccsdiff',0.03010905419334576 3),doc(sccsdiff,0.03010905419334576 3),doc(admin,0.02921791327876963),doc('sccs-admin',0.02921791327876963),doc(mv,0.02888346131045819),doc(sum,0.02888346131045819),doc(tftp,0.02877942535736114 5),doc(chmod,0.02877351016668773 5),doc(dd,0.02853613363413844),doc(vswap,0.02851815599313116 66),doc(more,0.02692124497796938),doc(page,0.02692124497796938),doc('sccs-unget',0.02688588474402535 4),doc(unget,0.02688588474402535 4),doc('old-ccat',0.02677621040931234 6),doc('old-compact',0.02677621040931234 6),doc('old-uncompact',0.02677621040931234 6),doc(find,0.02669059610320651),doc(sact,0.02654706287361875 7),do
```

```
c('sccs-
show',0.026547062873618757),doc(
indent,0.026119550017807504),doc
(bar,0.026097301314349095),doc(t
ar,0.026097301314349095),doc(ar,
0.025444251709980378),doc(file,0
.025373949651683568),doc(screend
ump,0.025213506362881864),doc(sc
reenload,0.025213506362881864),d
oc(cluster,0.024944546643102663)
,doc(diff,0.024919726714278643),
doc(touch,0.024630194606313736),
doc(cut,0.024549122358853374),do
c(head,0.024384320270116992),doc
('old-
filemerge',0.024341638289465141),
doc(checknr,0.024198152519025608
),doc(dos2unix,0.023660964700880
09),doc(unix,0.023660964700880009
),doc(ln,0.023324226737813876),d
oc(pg,0.023252721888386174),doc(
make,0.023035687700083428),doc(o
bjdump,0.022983157104947137),doc(
diffmk,0.022970687053691836),doc
(strip,0.022780205353617966),doc
(df,0.022516279181966102),doc(uu
decode,0.022335458469113157),doc
(uuencode,0.022335458469113157),
doc(ctags,0.021986937386946126),
doc(csplit,0.021767835512164095)
,doc(crontab,0.021656423691379358
),doc(paste,0.020467999032035913
),doc(error,0.020005295816877777
),doc(du,0.019926189370223088),d
oc(cmp,0.019904431066690403),doc
(sdiff,0.019281476074164181)]}).
query_doc_rel_th(reference(22),
docRel([doc(egrep,0.7400499677604623),doc(fgrep,0.74004996776046
23),doc(grep,0.7400499677604623)
,doc(acctcom,0.38496469472295125
),doc(which,0.3630546187240132)]
,doc(apropos,0.35775335635243405
),doc(whereis,0.3263466559000397
5),doc(awk,0.29814996802899363),
doc(nawk,0.29814996802899363),do
c(lptest,0.2691150289473864),doc
(test,0.25745723424415556),doc(e
xpr,0.24174902912398513),doc(cpi
o,0.14225634041878665),doc(paxcp
io,0.14225634041878665),doc(logi
n,0.14179320062295342),doc(look,
0.14082568216804944),doc(gfxtool
,0,107508998858848023),doc(shellt
ool,0.10210907101257512),doc(clo
ck,0.10193497972174951),doc(strin
gs,0.10171991677432513),doc(newg
rp,0.0989811010911113],doc(who,
0.09710233733333889),doc(lookbib,
0.09597741005593611),doc(wall,0.
09107295161066442),doc(users,0.0
9030313972344387),doc(getopt,0.08
898421852036931),doc(getoptcvt,0.
08898421852036931),doc(getopts,0.
08898421852036931),doc(ypcat,0.08
786201888199484),doc(yppasswd,0.
08721314369583379),doc(pr,0.0851
35946852372331),doc(banner,0.0848
4155528643915),doc(rev,0.0830418
5214790973),doc(comm,0.082198377
65784539),doc(inline,0.080282852
59042331),doc(rusers,0.080170393
98014841),doc(rwho,0.080170393998
014841),doc(logname,0.07673663237
040255),doc(lastcomm,0.07630005
651779373),doc(cp,0.069732840570
48762),doc(perfmeter,0.062819950
55841103),doc(w,0.060932574539931
03),doc(rm,0.05046173609757911)
,doc(rmdir,0.050461736097579111),
doc(mkstr,0.050205652969062841),d
oc(rcp,0.049584280192067884),doc
(install,0.047492752345664571),do
c(pack,0.046779652585835586),doc
(pcat,0.046779652585835586),doc(
unpack,0.046779652585835586),doc
(rmdel,0.044038530283961874),doc
('sccs-
rmdel',0.044038530283961874),doc
('sccs-
val',0.042456658285711751),doc(va
l,0.042456658285711751),doc(ftp,0
.042077761229791311),doc(delta,0.
041465078564629734),doc('sccs-
comb',0.041465078564629734),doc(
uusend,0.038158819800103941),doc(
size,0.037754779598159551),doc(cr
ypt,0.036890569362787451),doc(org
anizer,0.036320453647991511),doc(g
et,0.035442710416658786),doc('sc
s-
get',0.035442710416658786),doc(u
upick,0.034745811519927851),doc(u
uto,0.034745811519927851),doc(tai
l,0.034233618313092281),doc(rdist
,0.033719557879505771),doc(what,0
.033716083108296561),doc(prt,0.03
358373540148141),doc('sccs-
prt',0.033583735401481411),doc(spl
it,0.032132755367422536),doc(chg
rp,0.031940463924408954),doc(com
press,0.031817503785499766),doc(
uncompress,0.031817503785499766)
,doc(zcat,0.031817503785499766),
doc('sccs-
sccsdiff',0.031689065780982921),d
oc(sccsdiff,0.031689065780982921)
,doc(admin,0.030751161093549407)
,doc('sccs-
admin',0.030751161093549407),doc
(mv,0.030399158325334175),doc(su
m,0.030399158325334175),doc(tftp
,0.030289662950949148),doc(chmod
,0.030283437352989374),doc(dd,0.
030033604178905795),doc(vswap,0.
030014687283544266),doc(more,0.0
283339721511866281),doc(page,0.02
833397215118662 8),doc('sccs-
unget',0.028296756343201171),doc(
unget,0.028296756343201171),doc('
old-
ccat',0.028181326705827365),doc(
'old-
compact',0.028181326705827365),d
oc('old-
uncompact',0.028181326705827365)
,doc(find,0.028091219678201745),
doc(sact,0.027940154356622556),d
oc('sccs-
show',0.027940154356622556),doc(
indent,0.027490207210391281),doc(
bar,0.027466790977423355),doc(ta
r,0.027466790977423355),doc(ar,0
.026779471749851652),doc(file,0.
026705480500056811),doc(screendum
p,0.026536617742020461),doc(scree
nload,0.026536617742020461),doc(c
luster,0.026253544012842963),doc
(diff,0.026227421626130547),doc(
touch,0.025922695946071345),doc(
cut,0.025837369327497211),doc(hea
d,0.025663919035858195),doc('old
-
filemerge',0.025618997263030206)
,doc(checknr,0.025467981891079457
),doc(dos2unix,0.024902604446917
89),doc(unix,0.024902604446917789
),doc(ln,0.024548195723414455),
doc(pg,0.024472938564467413),doc
(make,0.024244515226235551),doc(o
bjdump,0.024189228020131664),doc
(diffmk,0.024176103586797046),do
c(strip,0.023975626112979611),doc
(df,0.023697850073882133),doc(uu
decode,0.023507540560093833),doc
(uuencode,0.023507540560093833),
doc(ctags,0.023140730383064384),
doc(csplit,0.022910130853829126)
,doc(crontab,0.022792872553553233
],doc(paste,0.021542083772084303
],doc(error,0.021055099606854068
],doc(du,0.020971841947027125),d
oc(cmp,0.020948941848355751),doc(
sdiff,0.020293296486333115)]}).
```

BIBLIOGRAFIA

AAIS 91. *AAIS Prolog Reference Manual,* Advanced A.I. Systems Prolog.

Aalbersberg, I. J., 1992. "Incremental Relevance Feedback", *Proceedings of SIGIR'92*, ACM Press.

Abramson H., Dahl, V., 1989. *Logic Grammars*, Springer-Verlag, New York.

Aho, A., Sethi, R., Ullman, J., 1985. *Compilers. Principles, Techniques, and Tools,* Addison-Wesley Publishing Company, England.

Akscyn, R., McCracken D., 1988. "KMS: A distributed Hypermedia System for managing knowledge in organizations", *Communications of the ACM*, Vol.31, No.7.

Allan, J., 1995. *Automatic Hypertext Construction.* Ph.D. Dissertation. Department of Computer Science, Cornell University. USA.

Allen, J., 1981. "Natural Language Understanding" en (ed.) Barr, A., Cohen, P.,*The Handbook of Artificial Intelligence, Vol IV*, Addison-Wesley.

Allen, J., 1991. "The RHET System", *ACM SIGART*, Vol.2, No.3.

Allen, J., 1994. *Natural Language Understanding*, Benjamin/Cummings, California.

Allen, J., Buren, P., 1971. *Chomsky: Selected Readings*, Oxford University Press.

Alonso, J., 1994. "Traducción automática" en *Procesamiento del Lenguaje Natural: fundamentos y aplicaciones.* Documentación del curso de verano de 1994 de la Universidad Nacional de Educación a Distancia, Avila.

Amsler, R., 1989. "Research Toward the Development of a Lexical Knowledge Base for Natural Language Processing", *Proceedings of SIGIR'89*, ACM Press.

Anick, P., et al., 1991. "Adressing the Requirements of a Dynamic Corporate Textual Information Base", *Proceedings of SIGIR'91*, ACM Press.

Antonacci, F., Russo, M., Pazienza, M., Velardi, P., 1989. "A system for text analysis and lexical knowledge adquisition", *Data and Knowledge Engineering Vol.4*, North Holland.

Arroyo, M. I., Gallego Martinez, M. A., 1993. *Interface de sistema de ayuda.* Proyecto de fin de carrera, E.U.I.T. de Telecomunicación, Universidad Politécnica de Madrid.

Balasubramanian, V., 1993. *State of the Art Review on Hypermedia Issues And Applications*. Technical Report, Graduate School of Management, Rutgers University, Newark, New Jersey

Ballard, B., Jones, M. 1987. "Computational Lingüistics" en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Barnett, J., Kevin K., 1990. "Knowledge and Natural Language Processing", *Communications of the ACM*, Vol.33, No.8.

Barr, A., Cohen, P., Feigenbaum, E.A., 1981. *The Handbook of Artificial Intelligence, Vol IV*, Addison-Wesley.

Barret, E., 1989. *The Society of text: Hypertext, Hypermedia and the social construction of information*, MIT Press.

Bartell, B., Cottrell, G., Belew, R., 1992. "Latent Semantic Indexing is an Optimal Special Case of Multidimensional Scaling", *Proceedings of SIGIR'92*, ACM Press.

Bates, M., Bobrow, R., 1983. "Information Retrieval Using a Transportable Natural Language Interface", *Proceedings of SIGIR'83*, ACM Press.

Bates, M., 1987. "Natural-Language Interfaces" en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Beardon, C., Lumsden, D., Holmes G., 1991. *Natural Language and Computational Linguistics*, Ellis Horwood, New York.

Belew, R., 1989. "Adaptive Information Retrieval: Using a connectionist representation to retrieve and learn about documents", *Proceedings of SIGIR'89*, ACM Press.

Belkin, N., Croft, B., 1992. "Information Filtering and Information Retrieval: Two Sides of the Same Coin?", *Communications of the ACM*, Vol.35, No.12.

Belkin, N., Croft, W., 1993. "The Effect of Multiple Query Representations on Information Retrieval System Performance", *Proceedings of SIGIR'93*, ACM Press.

Bennet, P., 1995. *A Course in Generalized Phrase Structure Grammar*, University College London Press.

Biggerstaff, T., Richter C., 1989a. "Reusability framework, assessment, and directions", en (ed.) Biggerstaff, T., Perlis, A., *Software Reusability, Volume I: Concepts and Models*, ACM Press.

Biggerstaff, T., Perlis, A., 1989b. *Software Reusability, Volume I: Concepts and Models*, ACM Press.

Binot, J., 1991. "Natural Language Processing and Logic", en (ed.) Thayse, A. *From Natural Language Processing to Logic for Expert Systems*, John Wiley, Sons, New York.

Birnbaum, L., Selfridge, M., 1981. "Conceptual Analysis of Natural Language", en (ed) Schank, R., Riesbeck, C., *Inside Computer Understanding*, Lawrence Erlbaum Associates Publishers, New Jersey.

Blair, D., Maron, M.1985. "An evaluation of retrieval effectiveness for a full-text document retrieval system", *Communications of ACM*, Vol.28, No.8.

Boguraev, B., et al., 1995. Editorial, *Natural Language Engineering*, Vol.1, No.1, Cambridge University Press.

Bollmann, P., 1983. "The Normalized Recall and Related Measures", *Proceedings of SIGIR'83*, ACM Press, Cambridge, MA.

Booch, G., 1991. *Object-Oriented Design with Applications*, Benjamin/Cummings.

Boy, G., 1991. *Intelligent Assistant Systems*, Academic Press, London.

Bruce, B., Moser M., 1987. "Case Grammars", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Buckley, C., 1985. *The Implementation of the SMART system*, Tech Report 85-686, Department of Computer Science, Cornell University, Ithaca NY.

Buenaga, M., Fernández Chamizo, C., 1992: *Procesamiento del Lenguaje Natural: Sistemas de Análisis y Comprensión*. Informe Técnico 92/9, Departamento de Informática y Automática, Universidad Complutense de Madrid.

Buenaga, M., Fernandez-Manjon, B., Vaquero, A., 1993a. "An On-Line Intelligent Assistant for UNIX", *Proceedings of Teleteaching 93*, Trondheim, Norway.

Buenaga, M., Fernández-Manjón, B., Vaquero, A., 1993b. "Un asistente inteligente para Unix basado en la documentación". *Revista de Enseñanza y Tecnología*, Vol.1, No.2.

Buenaga, M., Fernandez-Manjon, Fernandez-Valmayor, A., 1994. "Information Overload in the Information Age". *IFIP Joint Working Conference, WG 3.2 and WG 3.6, Adults in Innovative Learning Situations*, Nantes, France.

Buenaga, M., Fernández-Manjón, B., Fernández-Valmayor, A., 1995. "Information Overload in the Information Age", en (ed.) Collis, B., Davies, G., *Innovative Adult Learning with Innovative Technologies*, North-Holland.

Burton, R., 1987. "Semantic Grammar", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Can, F., Ozkarahan, E., 1983. "A Clustering Scheme", *Proceedings of SIGIR'83*, ACM Press, Cambridge, MA.

Carberry, S., Kobsa, A., 1992. "User Modeling and User-Adapted Interaction", *Tutorial of the Tenth National Conference on Artificial Intelligence* (AAAI-92).

Carbonell, J., Hayes, P., 1987. "Natural-Language Understanding", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Carroll, T., Grover, C., 1993. *The Alvey Natural Language Tools Grammar (4th Release)*, Human Communication Research Centre, University of Edinburgh, UK.

Castell, N. et al., 1993. "Control de las especificaciones de software escritas en lenguaje natural", *Actas de la V Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 93)*.

Chin, D. N., 1989. "KNOME: Modelling What the User Knows in UC". En A. Kobsa, W. Wahlster (Eds.), *User Models in Dialog Systems*, Sprinter-Verlag.

Clayton, J., Scott, C., 1992. "Knowledge Acquisition Techniques", *Tutorial of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San José, California.

Cleverdon, C., 1991. "The significance of the Cranfield Tests on Index Languages", *Proceedings of SIGIR'91*, ACM Press.

Clocksin, W., Mellish, C., 1987. *Programming in PROLOG*, Springer-Verlag, New York.

Coffin, S., 1995. *UNIX Sistema V versión 4 · Manual de Referencia*, McGraw-Hill.

Coll-Vinent, R., 1980. *Teoría de la Teledocumetación*, A.T.E., Barcelona.

Conklin, J. 1987. "Hypertext: An Introduction and Survey", *IEEE Computer*, Vol.20, No.9.

Cooper, W., 1991. "Some Inconsistencies and Misnomers in Probabilistic Information Retrieval", *Proceedings of SIGIR'91*, ACM Press.

Cortés, U., Béjar A., Moreno, A., 1993. *Inteligencia Artificial*, Ediciones UPC, Barcelona.

Croft, B. et al., 1991. "The Use of Phrases and Structured Queries in Information Retrieval", *Proceedings of SIGIR'91*, ACM Press.

Croft, B. et al., 1992a. "Experience with Large Document Collections", *Proceedings of SIGIR'92*, ACM Press.

Croft, B., Smith, L., Turtle, H., 1992b. "A Loosely-Coupled Integration of a Text Retrieval System and an Object-Oriented Database System", *Proceedings of SIGIR'92*, ACM Press.

Croft, B., 1993. "Knowledge-Based and Statistical Approaches to Text Retrieval", *IEEE Expert*, Vol.8, No.2.

Crouch, C., Yang B., 1992. "Experiments in Automatic Statistical Thesaurus Construction", *Proceedings of SIGIR'92*, ACM Press.

Cullingford, R., 1981. "SAM", en (ed) R. Schank, R., Riesbeck, C., *Inside Computer Understanding*, Lawrence Erlbaum Associates Publishers, New Jersey.

Cybulski, J., Reed K., 1992. "A Hypertext Based Software-Engineering Environment", *IEEE Software*, Vol.9, No.3.

Date, C., 1993. *Introducción a los sistemas de bases de datos, Vol.1*, Addison-Wesley.

Delsarte, P., Thayse A., 1991. "Extensions of Montague semantics", en (ed.) Thayse, A. *From Natural Language Processing to Logic for Expert Systems*, John Wiley & Sons, New York.

Devanbu, P., et al., 1991. "LaSSIE: A knowledge-based software information system", en Lowry, M., McCartney, R., (eds.) *Automatic Software Design*, AAAI Press/ MIT Press, Cambridge.

Díaz, A., de Pedro, J., Buenaga, M., 1995. *Diseño e implementación de un sistema de selección de componentes software basado en técnicas de indexación automática de texto*, Informe Técnico 95-15, Departamento de Informática y Automática, Univesidad Complutense de Madrid.

Digitalk 93. *Smalltalk V, Object Oriented Programming System for Macintosh*, Digitalk Incorporated, L.A., California.

Downton, A., 1991. *Engineering The Human-Computer Interface*. Edit. Mac-Graw Hill.

Dyer, M., 1983. *In Depth Understanding*, MIT Press, Cambridge.

Edwards, J., Lampert, M., 1992. *Talking Data: Transcription and Coding in Discourse Research*, Lawrence Erlbaum Associates Publishers, New Jersey.

Evans, D, Kimberly, G., 1991. "Automatic Indexing Using Selective NLP and First-Order Thesauri", *Proceedings of RIAO'91*, Barcelona.

Evans, D., Carbonell, J., 1993. "Explorations of NLP for Information Management: Observations from Practice in Mono- and Multi- lingual Applications", *Proceedings of SIGIR'93*, ACM Press.

Fagan, J., 1987. *Experiments in Automatic Phrase Indexing for Document Retrieval: A comparison of syntactic and non-syntactic methods* Ph.D. thesis, Department of Computer Science, Cornell University.

Faloutsos, C., 1985. "Access Methods for Text", *ACM Computing Surveys*, Vol.17, No.1

Fernández-Chamizo, C., et al., 1993. "A Case-Based approach to Software Component Retrieval", en *Case-Based Reasoning and Information Retrieval*, AAAI Press, Menlo Park, CA.

Fernández-Chamizo, C. et al., 1995. "Case-Based Retrieval of Software Components", *Experts Systems With Applications*, Vol 9., No.1.

Fernández-Manjon, B., Buenaga, M, 1993. "Argos: An On Line Intelligent Assistant", *World Conference on Educational Multimedia and Hypermedia*, Orlando, USA.

Fernández-Manjon, B., Buenaga, M., Fernández-Valmayor, A., 1994. "De los entornos de ayuda inteligentes a los tutores inteligentes: el sistema Argos", *II Congreso Ibero-Americano de Informática na Educaçao*, Lisboa, Portugal.

Fernández-Manjon, B., Buenaga, M., Fernández-Valmayor, A., 1995a. "The integration of technologies and the reusing of information in IHS", *World Conference on Educational Multimedia and Hypermedia*, Graz, Austria.

Fernández-Manjon, B., Buenaga, M., Fernández-Valmayor, A., 1995b. "IHS: An Engineering Step Towards ITS". *IFIP World Conference of Computers in Education*, Birmingham, UK.

Fernández-Manjon, B., Buenaga, M., 1995c. "Internet como herramienta de trabajo en el campo educativo". *Revista de Enseñanza y Tecnología*, Vol.4, No.1.

Fernández-Valmayor Crespo, A., 1990. *Diseño de una base de conocimientos dinámica y su aplicación en un entorno educativo*, tesis doctoral, Departamento de Informática y Automática, Universidad Complutense. Editorial de la Universidad Complutense de Madrid.

Fernández-Valmayor, A., Fernández, C., 1992. "Educational and Research Utilization of a Dynamic Knowledge Base", en (ed.) Kibby, M., Hartley, J., *Computer Assisted Learning*, Pergamon Press.

Fernández-Valmayor, A., Villarrubia, C., Buenaga, M., 1993a. "An Intelligent Interface to a Database System", en *Case-Based Reasoning and Information Retrieval*, AAAI Press, Menlo Park, CA.

Fernández-Valmayor, A., Villarrubia, C., Buenaga, M., 1993b. "An Intelligent Interface to a Database System", *Symposium on Case-Based Reasoning and Information Retrieval*, AAAI Spring Symposium Series.

Fischer, G., 1989. "Human-Computer Interaction Software: Lesson Learned, Challenges Ahead", *IEEE Software*, Vol.6, No.1.

Foltz, P., 1991. *Models of Human Memory and Computer Information Retrieval: Similar Approaches to Similar Problems*, ICS Tech Report 91-03, Department of Psychology and Institute of Cognitive Science, University of Colorado.

Fowler, R., Fowler, W., Wilson, B., 1991. "Integrating Query, Thesaurus, and Documents Through a Common Visual Representation", *Proceedings of SIGIR'91*, ACM Press, Cambridge, MA.

Fox, C., 1992a. "Lexical Analysis and Stoplists" en (ed.) Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Fox, C., 1992b. "Extended Boolean Models" en (ed.) Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Frakes, W., Gandel, P., 1989. "Information Retrieval and Software Reuse", *Proceedings of SIGIR'89*, ACM Press, Cambridge, MA.

Frakes, W., Baeza, R., 1992a. *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Frakes, W., 1992b. "Stemming Algorithms" en (ed.) Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Frakes, W., Pole, T., 1994. "An Empirical Study of Representation Methods for Reusable Software Components". *IEEE Trans. Software Engineering*, Vol.20, No.8.

Friedman C., et al., 1995. "Natural language processing in an operational clinical information system", *Natural Language Engineering*, Vol.1, No.1., Cambridge University Press.

Frost, R., 1986: *Introduction to Knowledge Base Systems*, Ed. Collins, London, England.

Frost, R., 1989: *Bases de datos y sistemas expertos: Ingeniería del conocimiento*, Ediciones Díaz de Santos, Madrid.

Gazdar, G, Mellish, C., 1989. *Natural Language Processing in Prolog*, Addison-Wesley Publishing Company, England.

Gómez, F, Segani, C., 1991. "Classification Based Reasoning", *IEE Transactions on Systems, Man and Cybernetics*, Vol.21, No.3.

González, P., 1995. *Un enfoque basado en conocimiento de la síntesis de programas a partir de componentes reutilizables*, tesis doctoral (pendiente de presentación), Departamento de Informática y Automática, Universidad Complutense.

Gray, P., Lucas, R., 1988. *Prolog and Databases*, Ellis Horwood Limited, Chichester, England.

Grefenstette, G., 1992. "Use of Syntactic Context to Produce Term Association Lists for Text Retrieval", (ed.) Belkin, N., Ingwersen, P., *Proceedings of SIGIR'92*, ACM Press, Cambridge, MA.

Guha, R.V., Lenat, D.B., 1994. "Enabling Agents to Work Together", *Communications of the ACM*, Vol.37, No.7.

Guindon, R., 1988. "How to Interface to Advisory Systems? User Request Help with a Very Simple Language", (ed.) Soloway, E., Frye D., *Proceedings CHI'88*, ACM Press, New York.

Gurbaxani, V., Seungjin W., 1991. "The Impact of Information Systems on Organizations and Markets", *Communications of the ACM*, Vol.34, No.1

Hahn, H., 1995. *Internet. Manual de Referencia*, Ed. Mc-Graw-Hill.

Hardt, S., 1987. "Conceptual Dependency" en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Hardy, D., Schwartz, M., 1994. *Customized Information Extraction as a Basis for Resource Discovery*, Department of Computer Science, University of Colorado, Tech. Rep. CU-CS-707-94.

Harman, D., 1992a. "Relevance Feedback Revisited", *Proceedings of SIGIR'92*, ACM Press.

Harman D., 1992b. "Ranking Algorithms" en (eds.) Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Harman, D., 1992c. "Relevance Feedback and Other Query Modification Techniques", en Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Harman, D., 1993. "Overview of the First TREC Conference", *Proceedings of SIGIR'93*, ACM Press.

Hearst, M., 1994. *Context and Structure in Automated Full-Text Information Access* Ph.D. thesis, Computer Science Division, University of California at Berkeley.

Helm, R., Maarek, Y., 1991. "Integrating Information Retrieval and Domain Specific Aproaches for Browsing and Retrieval in Object Oriented Class Libraries", *Proceedings of the Conference on Object-Oriented Programming Systems, Languages, and Applications*, ACM Press.

Hess, M., 1992. "An Incrementally Extensible Document Retrieval System Based on Linguistic and Logical Principles", *Proceedings of SIGIR'92*, ACM Press.

Hirschheim, R., Heinz K., 1989. "Four Paradigms of Information Systems Development", *Communications of the ACM*, Vol.32, No.10.

Hobbs, J., Rau, L., 1992. "Text Interpretation", *Tutorial of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San José, California.

Holland, J., 1992. "Genetic Algorithms", *Scientific American*, July 1992.

Hopcroft, J., Ullman, J., 1979. *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley.

Hull, D., 1993. "Using Statistical Testing in the evaluation of Retrieval Experiments", *Proceedings of SIGIR'93*, ACM Press.

Iglesias, C. et al., 1993. "Procesamiento semántico en la arquitectura ARIES", *Actas de la V Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA 93).*

Jacobs, S., 1988. "Natural Language Techniques for Intelligent Information Retrieval", *Proceedings of SIGIR'88*, ACM Press.

Jacobs, S., Rau, F., 1990. "SCISOR: extracting information from on-line news", *Communications of the ACM*, Vol.33, No.10.

Johnson, E., Reichard, K., 1989. *X Window Applications Programming*, MIS Press.

Johnson, W., et al., 1968. "Automatic generation of efficient lexical processors using Finite States Techniques", *Communications of the ACM*, Vol.11, No.12.

Justeson, J., Katz, S., 1995. "Technical terminology: some linguistic properties and an algorithm for identification in text", *Natural Language Engineering*, Vol.1, No.1., Cambridge University Press.

Katalagarianos, P., Vassiliou, Y., 1995. "On the Reuse of Software: A Case-Based Approach Employing a Repository", *Automated Software Engineering*, Vol.2, No.1.

Kearsley, G., 1987. "Computer-Aided Instruction, Intelligent", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Kernighan, B., Ritchie, D., 1988. *C Programming Language*, Prentice-Hall.

Knuth, D., 1968. "Semantics of context-free languages", *Mathematical Systems Theory*, Vol.2, No.2.

Kobsa, W., Wahlster 1989. *User Models in Dialog Systems*, Sprinter-Verlag, Berlin.

Kobsa, A., 1991. "Fist experiences with the SB-ONE Knowledge Representation Workbench in NL Aplications", *ACM SIGART*, Vol.2, No.3.

Kobsa, A., 1992. "User Modeling and User-Adapted Interaction", *Tutorial of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San José, California.

Koskenniemi, K., 1987. "Morphology", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Kramer B., Mylopoulos J., 1987. "Knowledge Representation", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Krol, E., 1993. *The Whole Internet User's Guide and Catalog*. O'Reilly.

Krovetz, R., 1989a. "The Lexicon and Information Retrieval", (panel session), *Proceedings of SIGIR'89*, ACM Press.

Krovetz, R., Croft, W., 1989b. "Word Sense Disambiguation Using Machine-Readable Dictionaries", *Proceedings of SIGIR'89*, ACM Press.

Krovetz, R., 1992. "Corpus Linguistics and Information Retrieval", (panel session), *Proceedings of SIGIR'92*, ACM Press.

Krovetz, R., 1993. "Viewing Morphology as an Inference Process", *Proceedings of SIGIR'93*, ACM Press.

Krueger, C., 1992. "Software Reuse", *ACM Computing Surveys*, Vol.24, No.2

Kupiec, J., 1993. "MURAX: A Robust Linguistic Approach For Question Answering Using an On-Line Encyclopedia", *Proceedings of SIGIR'93*, ACM Press.

Kwok, K., 1989. "A Neural Network for Probabilistic Information Retrieval", *Proceedings of SIGIR'89*, ACM Press.

Lebowitz, M., 1983. "Intelligent Information Systems", *Proceedings of SIGIR'83*, ACM Press.

Lebowitz, M., 1987. "Memory Organization Packets", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Lebowitz, M., 1988. "The use of memory in text processing", *Communications of the ACM*, Vol.31, No.12

Lehnert, W.G., 1987, "Story Analysis", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Lehnert, W., Sundheim, B., 1991. "An Evaluation of Text Analysis Technologies", *AI magazine*, Vol.12, No.3.

Lenat, D., Prakash, M., Shepherd, M., 1986. "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks", *AI Magazine*, Vol.6, No.4.

Lenat, D., Guha, R., 1990a. "CYC: towards programs with Common Sense", *Communications of the ACM*, Vol.33, No.8.

Lenat, D., Guha, R., 1990b. *Building Large Knowledge-Based Systems - Representation and Inference in the Cyc Project*, Edit. Addison-Wesley.

Lesk, M., 1991. "SIGIR'91: The More Things Change, the More They Stay The Same", *ACM SIGIR Forum*, Vol.25, No.2.

Lewis, D., 1992. *Representation and Learning in Information Retrieval*. Ph.D. thesis, Department of Computer and Information Science, University of Massachusetts.

Li, X., Xing, D., 1992. "General Natural Language for Operating Systems", *ACM SIGART*, Vol.3, No.4.

Loeb, S., Terry, D., 1992. "Information Filtering", *Communications of the ACM*, Vol.35, No.12.

López-Muñiz, M., 1984. *Informática Jurídica Documental*, Díaz de Santos, Madrid.

Lowry, M., McCartney, R., 1991. *Automating Software Design*, AAAI Press/The MIT Press, Menlo Park.

Lytinen, S., 1992. "A Unification-Based, Integrated Natural Language Processing System", *Computers Math. Applic.*, Vol.23, No.6-9.

Maarek, Y., Berry, D., 1991. "An Information Retrieval Approach For Automatically Constructing Software Libraries", *IEEE Trans. Software Engineering*, Vol.17, No.8.

Maarek, Y., 1991. "Software Library Construction from an IR Perspective", *ACM SIGIR Forum*, Vol.25, No.2.

Maddix, F., 1990. *Human-Computer Interaction*, Ellis Horwood, London.

Maida, A., 1987. "Frame theory", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Martin, C., 1989. "Case-based parsing", en (ed.) Riesbeck, C., Schank, R., *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates Publishers, New Jersey.

Martin, J., 1992. "Computer Understanding of Metaphoric Language", *Cognitive Science*, Vol.16, No.2.

Mauldin, M., 1991. "Retrieval Performance in Ferret. A conceptual Information Retrieval System", *Proceedings of SIGIR'91*, ACM Press, Cambridge, MA.

McCord, M., 1987. "Natural Language Processing in Prolog" en (ed.) Walker, A et al., *Knowledge Systems and PROLOG*, Edit. Addison Wesley.

McCune, B. et al., 1985. "RUBRIC: A System for Rule Based Information Retrieval", *IEEE Transactions on Software Engineering*.

McDonald, D., 1987. "Natural-Language Generation", en (ed.) *Shapiro, S., Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Metzler, D., Haas, S., 1989. "The Constituent Object Parser: Syntactic Structure Matching for Information Retrieval", *Proceedings of SIGIR'89*, ACM Press.

Michalski, R., Kodratoff, Y., 1990. *Machine Learning. An Artificial Intelligence Approach. Volume III*, Morgan Kaufmann, California.

Mili, H., et al., 1995. "Resusing Software: Issues and Research Directions", *IEEE Transactions on Software Engineering*, Vol.21, No.6.

Miller, G. A. et al., 1993. *Five Papers on WordNet*, Cognitive Science Laboratory, Princeton University, CSL Report 43.

Nielsen, J., 1990. "The Art of Navigating through Hypertext", *Communications of the ACM*, Vol.33, No.3.

Noor, M., McGregor, D., 1992. *Natural Language Interfaces to Databases: State of the Art*, Tech. Report, Department of Computer Science, University of Strathclyde, Glasgow, UK.

Parikh, R., 1987. "Modal Logic", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Parc, 94. *VisualWorks Cookbook*, ParcPlace Systems, Sunnyvale, CA.

Pazzani, M., 1990. *Creating a Memory of Causal Relationships*, Lawrence Erlbaum Associates Publishers, New Jersey.

Pereira, F., Warren, H., 1980. "Definite clause grammars for language analysis, a survey of the formalism and a comparison with augmented transition network", *Artificial Intelligence* No.13.

Press, L., 1992. "The Net: Progress and Oportunity", *Communications of the ACM*, Vol.35, No.12.

Pressman, R., 1993. *Ingeniería del Software. Un efoque práctico*, McGraw-Hill.

Prieto-Díaz, R., Freeman, P., 1987. Classifying Software for Reusability, *IEEE Software*, Vol.4, No.1

Prieto-Díaz, R., 1989a. "Classification and Software Reusability: Research Issues" (panel), *Proceedings of SIGIR'89*, ACM Press, Cambridge, MA.

Prieto-Díaz, R., 1989b. "Classification of Reusable Modules", en (ed.) Biggerstaff, T., Perlis, A., *Software Reusability, Volume I: Concepts and Models*, ACM Press.

Prieto-Díaz, R., 1990. "Domain Analysis: An introduction", *ACM SIGSOFT*, Vol.15, No.2.

Prieto-Diaz, R., 1991. "Implementing Faceted Classification for Software Reuse", *Communications of the ACM*, Vol.34, No.5.

Qiu, Y., Frei, H., 1993. "Concept Based Query Expansion", *Proceedings of SIGIR'93*, ACM Press.

Raghavan, V., Bollmann, P., Jung, G., 1989. "Retrieval System Evaluation Using Recall and Precision: Problems and Answers", *Proceedings of SIGIR'89*, ACM Press.

Ram, A., 1992. "Natural Language Understanding for Information-Filtering Systems", *Communications of the ACM*, Vol.35, No.12.

Rasmussen E., 1992. "Clustering Algorithms" en (eds.) Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Rau, L., Jacobs, P., 1991. "Creating Segmented Databases From Free Text for Text Retrieval", *Proceedings of SIGIR'91*, ACM Press.

Rich, C., Waters, R. C., 1986. *Readings in Artificial Intelligence and Software Engineering*. Morgan Kaufmann Publishers, Inc., Los Altos, California.

Rich, E., Knight K., 1991. *Artificial Intelligence*, McGraw-Hill, New York.

Riesbeck, C, Schank, R., 1989. *Inside Case-Based Reasoning*, Lawrence Erlbaum Associates Publishers, New Jersey.

Rijsbergen, C., 1979. *Information Retrieval*, Butterworths.

Rijsbergen, C., 1989. "Towards an Information Logic", *Proceedings of SIGIR'89*, ACM Press.

Ritchie, A., Thanisch, P., 1995. "Natural language interfaces to databases - an introduction", *Natural Language Engineering*, Vol.1, No.1., Cambridge University Press.

Rodríguez, H., 1994. "Sintáxis, Semántica y Pragmática" en *Procesamiento del Lenguaje Natural: fundamentos y aplicaciones*. Documentación del curso de verano de 1994 de la Universidad Nacional de Educación a Distancia, Avila.

Rowe, N., 1988. *Artificial Intelligence through PROLOG*, Edit Prentice Hall.

Rumbaugh, J., 1991. *Object-oriented modeling and design*, Prentice-Hall, New Jersey.

Salton, G., Wong, A., Yang, C., 1975. "A Vector Space Model for Automatic Indexing", *Communications of the ACM*, Vol.18, No.11.

Salton, G., McGill, M.J., 1983a. *Introduction to Modern Information Retrieval*, McGraw-Hill, New York.

Salton, G, Fox, E.A., Wu, H., 1983b. "Extended boolean Information Retrieval", *Communications of the ACM*, Vol.26, No.12.

Salton, G., 1983c. "Research Problems in Information Retrieval", *Proceedings of SIGIR'83*, ACM Press.

Salton, G., 1986. "Another Look at Automatic Text-Retrieval Systems", *Communications of the ACM*, Vol.29, No.7.

Salton, G., 1989a. *Automatic Text Processing: the transformation, analysis and retrieval of information by computer*, Edit. Addison Wesley.

Salton, G., Smith, M., 1989b. "On the application of syntactic methodologies in authomatic text analysis", *Proceedings of SIGIR'89*, ACM Press.

Salton, G., 1991a. "The SMART Document Retrieval Project", *Proceedings of SIGIR'91*, ACM Press.

Salton, G., 1991b. *The State of Retrieval System Evaluation*, Tech Report 91-1206, Department of Computer Science, Cornell University, Ithaca NY.

Salton, G., Allan J., 1993. "Approaches to Passage Retrieval in Full Text Information Systems" *Proceedings of SIGIR'93*, ACM Press.

Scha, R., Bruce B., Polanyi, L., 1987. "Discourse Understanding" en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, New York.

Schank, R., 1972. "Conceptual Dependency: A Theory of Natural Language Understanding" en *Cognitive Psichology*, No.3.

Schank, R., Abelson, R., 1977. *Scripts, Plans, Goals and Understanding*, Lawrence Erlbaum Associates Publishers, New Jersey.

Schank, R, Riesbeck, C., 1981. *Inside Computer Understanding*, Lawrence Erlbaum Associates Publishers, New Jersey.

Schank, R., 1982. *Dynamic Memory: A theory of Reminding and Learning in Computers and People*, Cambridge University Press, New York.

Schwartz, C., 1993. "Information Science Abstracts", *American Society of Information Science (ASIS)*, Vol.28, No.9.

Shieber, S., 1989. *Introducción a los formalismos gramaticales de unificación*, Editorial Teide, Barcelona.

Shieber, S., 1992. *Constraint-Based Grammar Formalisms*, MIT Press, Cambridge, MA.

Sicstus 93., *SICStis Prolog User's Manual*, Swedish Institute of Computer Science.

Silberschatz, A. et al., 1991. "Database Systems: Achievements and Opportunities", *Communications of the ACM*, Vol.34, No.10.

Smeaton, A., 1988. "Experiments on incorporating Syntactic Processing of User Queries into a Document Retrieval Strategy", *Proceedings of SIGIR'88*, ACM Press.

Smith, J., Weiss, S., 1988. "Hypertext", *Communications of the ACM*, Vol.31, No.7.

Smith, P., Shute, S., Galdes, D., 1989. "In Search of Knowledge-Based Search Tactics", *Proceedings of SIGIR'89*, ACM Press, Cambridge, MA.

Sonnenwald, D., 1992. "Developing A Theory to Guide the Process of Designing Information Retrieval Systems", *Proceedings of SIGIR'92*, ACM Press.

Sopeña, L., 1983. "Natural Language Grammars for an Information System", *Proceedings of SIGIR'83*, ACM Press.

Sparck, K., 1991. "Notes and references on early automatic classification work", *ACM SIGIR Forum*, Vol.25, No.1.

Srinivasdan, P., 1992. "Thesaurus Construction" en Frakes, W., Baeza, R., *Information Retrieval: Data Structures and Algorithms*, Prentice Hall, London.

Sun 1988. *SunOS Programmer's Guide Vol.2, Programming Utilities, Libraries*. Sun Microsystems Europe.

Sun 1989. *SunOs Reference Manual v.4.1.* Sun Microsystems Europe.

Thayse, A., 1991. *From Natural Language Processing to Logic for Expert Systems*, John Wiley, Sons, New York.

TMC 1991. *WAIStation reference manual v.0.57*, Thinking Machines Corporation, Cambridge, Massachusetts.

Tomita, M., 1986. *Efficient Parsing for Natural Language.* Kluwer Academic Publisher.

Tong, R., et al., 1987. "Conceptual Information Retrieval using RUBRIC", *Proceedings of SIGIR'87*, ACM Press.

Ullman, J., 1988. *Principles of Database and Knowledge-Base Systems, Vol.I*, Computer Science Press.

Verdejo, F., 1985. "Comprensión automática del Lenguaje Natural", *Mundo Electrónico*, No.150.

Verdejo, F., 1994. "Comprensión del Lenguaje Natural: avances, aplicaciones y tendencias" en *Procesamiento del Lenguaje Natural: fundamentos y aplicaciones.* Documentación del curso de verano de 1994 de la Universidad Nacional de Educación a Distancia, Avila.

Villarrubia, C., Fernández-Valmayor, A., Fernández-Chamizo, C., 1992. "Intelligent Retrieval of Information in a Database System", *IFIP World Computer Congress*, Madrid.

Voorhees, E., 1993. "Using WordNet to Disambiguate Word Senses for Text Retrieval", *Proceedings of SIGIR'93*, ACM Press.

Walker, A, McCord, M, Sowa, J, Wilson, W., 1987. *Knowledge Systems and PROLOG*, Addison Wesley.

Watt, D., 1991. *Programming Language Syntax and Semantics*, Prentice Hall, London.

Wendlandt, E., Driscoll, J., 1991. "Incorporating a Semantic Analysis into a Document Retrieval Strategy", *Proceedings of SIGIR'91*, ACM Press.

Wenger, E., 1987. *Artificial Intelligence and Tutoring Systems*, Morgan Kaufman Publishers Inc., Los Altos, CA.

White, G., 1990. "Natural Language Understanding and Speech Recognition", *Communications of the ACM*, Vol.33, No.8.

Wilensky, R., 1981. "PAM", en (ed) R. Schank, R, Riesbeck, C., *Inside Computer Understanding*, Lawrence Erlbaum Associates Publishers, New Jersey.

Wilensky, R., Chin, D. N., et al., 1989. *The Berkeley UNIX Consultant Project*. Informe técnico UCB//CSD-89-520. University of California, Berkeley. USA.

Wilkinson, R., Hingston, P., 1991. "Using the Cosine Measure in a Neural Network for Document Retrieval", *Proceedings of SIGIR'91*, ACM Press.

Williams, M., 1993. "The State of Databases Today: 1993", en Young, K. (ed.) *Gale Directory of Databases*, Gale Research Inc, Detroit.

Winograd, T., 1983. *Languaje as a cognitive Process Vol.1. Syntax*, Addison Wesley Publishing Company, Reading, Massachusetts.

Winston, P., 1984a. *Artificial Intelligence*, Addison Wesley Publishing Company, Reading MA.

Winston, P, Horn, B., 1984b. *LISP*, Addison Wesley Publishing Company, Reading MA.

WN, 1993. *WordNet Reference Manual*, WordNetTM, Cognitive Science Laboratory, Princeton University.

Wood, M., Sommerville, I., 1988. "An Information Retrieval System for Software Components", *ACM SIGIR Forum*, Vol.22, Nos. 3/4.

Woods, W., 1987. "Grammars. Augmented Transition Network", en (ed.) Shapiro, S., *Encyclopaedia of Artificial Intelligence*, Edit. Willey, NewYork.

Young, K., 1993. *Gale Directory of Databases*, Gale Research Inc, Detroit.

Yourdon, E., Constantine, L. 1979. *Structured Design*, Yourdon Press (Prentice Hall).

Yu, C.T., Luk, W.S., Cheung T.Y., 1976, "A Statistical Model for Relevance Feedback in Information Retrieval", *Journal of the ACM*, April 1976.